
5 PTAS.
(IVA Incluido)
103
mi computer
CURSO PRACTICO DEL ORDENADOR PERSONAL,
EL MICRO Y EL MINIORDENADOR
File Edit Create Font Size Style
1
2
3
4
1
2
3
4
Editorial Delta, S.A.

MI COMPUTER CURSO PRACTICO
DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Publicado por Editorial Delta, S.A., Barcelona

Volumen IX-Fascículo 103

Director: José Mas Godayol
Director editorial: Gerardo Romero
Jefe de redacción: Pablo Parra
Coordinación editorial: Jaime Mardones, Francisco Martín
Asesor técnico: Ramón Cervelló

Redactores y colaboradores: G. Jefferson, R. Ford, F. Martín, S. Tarditti, A. Cuevas, F. Blasco
Para la edición inglesa: R. Pawson (editor), D. Tebbutt (consultant editor), C. Cooper (executive editor), D. Whelan (art editor), Bunch Partworks Ltd. (proyecto y realización)

Realización gráfica: Luis F. Balaguer

Redacción y administración:
Aribau, 185, 1.°, 08021 Barcelona
Tel. (93) 209 80 22 - Télex: 93392 EPPA

MI COMPUTER, *Curso práctico del ordenador personal, el micro y el miniordenador*, se publica en forma de 120 fascículos de aparición semanal, encuadernables en diez volúmenes. Cada fascículo consta de 20 páginas interiores y sus correspondientes cubiertas. Con el fascículo que completa cada uno de los volúmenes, se ponen a la venta las tapas para su encuadernación.
El editor se reserva el derecho de modificar el precio de venta del fascículo en el transcurso de la obra, si las circunstancias del mercado así lo exigieran.

ISBN: 84-85822-83-8 (fascículo) 84-7598-181-X (tomo 9)
84-85822-82-X (obra completa)
Depósito Legal: B. 52-84

Fotocomposición: Tecfa, S.A., Pedro IV, 160, Barcelona-5
Impresión: Cayfosa, Santa Perpètua de Mogoda (Barcelona) 078601
Impreso en España-Printed in Spain-Enero 1986

Editorial Delta, S.A., garantiza la publicación de todos los fascículos que componen esta obra.

Distribuye para España: Marco Ibérica, Distribución de Ediciones, S.A., Carretera de Irún, km 13,350. Variante de Fuencarral, 28034 Madrid.
Distribuye para Colombia: Distribuidoras Unidas, Ltda., Transversal 93, n.° 52-03, Bogotá D.E.
Distribuye para México: Distribuidora Intermex, S.A., Lucio blanco, n.° 435, Col. San Juan Tlihuaca, Azcapotzalco, 02400, México D.F.
Distribuye para Venezuela: Distribuidora Continental, S.A., Edificio Bloque Dearmas, final Avda. San Martín con final Avda. La Paz, Caracas 1010.

Pida a su proveedor habitual que le reserve un ejemplar de MI COMPUTER. Comprando su fascículo todas las semanas y en el mismo quiosco o librería, Vd. conseguirá un servicio más rápido, pues nos permite realizar la distribución a los puntos de venta con la mayor precisión.

Servicio de suscripciones y atrasados (sólo para España)

Las condiciones de suscripción a la obra completa (120 fascículos más las tapas, guardas y transferibles para la confección de los 10 volúmenes) son las siguientes:

a) Un pago único anticipado de 27 105 ptas. o bien 10 pagos trimestrales anticipados y consecutivos de 2 711 ptas. (sin gastos de envío).

b) Los pagos pueden hacerse efectivos mediante ingreso en la cuenta 6.850.277 de la Caja Postal de Ahorros y remitiendo a continuación el resguardo o su fotocopia a Editorial Delta, S. A. (Aribau, 185, 1.°, 08021 Barcelona), o también con talón bancario remitido a la misma dirección.

c) Se realizará un envío cada 12 semanas, compuesto de 12 fascículos y las tapas para encuadernarlos.

Los fascículos atrasados pueden adquirirse en el quiosco o librería habitual. También pueden recibirse por correo, con incremento del coste de envío, haciendo llegar su importe a Editorial Delta, S.A., en la forma establecida en el apartado b).

No se efectúan envíos contra reembolso.



AVISO A LOS LECTORES

Como consecuencia de la aplicación del I.V.A., tras el ingreso de nuestro país en el Mercado Común, y de los continuos incrementos en los costes, nos vemos obligados —muy a pesar nuestro— a aumentar el precio de venta en el mercado español de los fascículos que integran esta obra.

Desearíamos manifestar nuestro agradecimiento a cuantos lectores nos honran con su confianza, pues estamos seguros de que sabrán comprender lo inevitable de esta medida.



IMPACTO DE UNA NUEVA TECNOLOGÍA

Hoy, muchos periódicos continúan componiendo tipográficamente sus páginas mediante una máquina denominada linotipia. Inventada en 1886, permite obtener líneas completas de matrices en metal fundido a partir de un texto entrado por teclado. La prensa de provincias que se encuentra en dificultades financieras está implantando la composición tipográfica informatizada a fin de reducir los costos. Gracias a los nuevos dispositivos es posible cargar el texto electrónicamente, obteniéndose placas de impresión de plástico llamadas «estereotipos»

Iniciamos una breve serie dedicada a las aplicaciones de la tecnología electrónica en la industria editorial. Analizando los métodos actuales de producción y comprendiendo los aspectos implicados, podremos hacernos una idea más cabal del revolucionario e irreversible cambio que está ejerciendo la informática en la edición de publicaciones periódicas.

Ya casi constituye un lugar común afirmar que la tecnología informática está modificando de manera radical la forma de vivir y de trabajar de la sociedad contemporánea. En efecto, la introducción de nuevos procedimientos basados en ordenador está ocasionando la desaparición de muchas prácticas y técnicas establecidas. La industria editorial ilustra claramente este cambio en las pautas de trabajo, dado que en este caso la tecnología electrónica se está introduciendo en una actividad que, en muchos casos, no ha actualizado sus métodos y maquinaria desde los años treinta. Además, la mano de obra de la industria gráfica está organizada en gremios tradicionalmente muy poderosos. Por tanto, es probable que la introducción de los ordenadores sea una experiencia traumática para quienes trabajan en esta actividad. En esta breve serie nos referiremos a los cambios que se están produciendo en la industria editorial y hacia dónde conducirán probablemente las actuales tendencias. En primer lugar, analicemos la nueva tecnología y veamos cómo se aplica en cada etapa del proceso editorial.

Comencemos por el ámbito más clásico, el de la palabra escrita. En la actualidad la mayoría de los escritores han reemplazado las máquinas de escribir mecánicas (incluso eléctricas) por microordenadores, que les han permitido escribir y corregir sus propios textos, incrementar su velocidad de trabajo y producir copias cuidadas y económicas para las editoriales. Es posible que esté familiarizado con programas de tratamiento de textos en disco y cinta o en ROMs especiales que se pueden instalar en su micro, y que le proporcionan capacidades para tratamiento de textos. De hecho, la aplicación más utilizada en micros personales, después de los juegos, es el tratamiento de textos.

Tanto como si desea enviarles un boletín interno a los socios de un club local como si quiere escribir su primera novela, es probable que el modesto micro personal sea una poderosa herramienta para su objetivo. Las copias se pue-

den comprobar y corregir en pantalla, estableciendo la anchura de los márgenes, numeración automática de páginas, títulos de página y pie antes de la impresión, para producir un único borrador ya acabado del texto. Para facilitar aún más las cosas, muchos paquetes de tratamiento de textos poseen verificadores de ortografía que van explorando el texto y señalando las palabras no reconocidas con el fin de que se las corrija. Si el texto ha de ser el cuerpo de un artículo en una revista o diario, la producción de un original pulcramente mecanografiado es sólo la primera etapa del largo proceso de producir una versión final impresa.

En la mayoría de los actuales métodos de producción de revistas aún prevalecen los sistemas tradicionales. Por lo general, la publicación espera que el autor del trabajo entregue el artículo mecanografiado (normalmente con máquina de escribir corriente o con un sistema de tratamiento de textos/impresora). El personal de la redacción de la publicación se encarga luego de revisar el artículo. El redactor lee y corrige el *contenido* del trabajo, normalmente escribiendo en él a mano o, cuando se requieren modificaciones sustanciales, produciendo hojas adicionales mecanografiadas. El texto pasa luego a un corrector, quien enmienda los errores ortográficos y gramaticales que se hubieran deslizado (nuevamente, por lo general, escribiendo a mano), reescribe, si fuera preciso, algunos párrafos para adecuar el texto al estilo de la publicación y prepara tipográficamente el original. El proceso de *preparación tipográfica* tiene como finalidad esencial indicar al tipógrafo el tipo de letra, el ancho de los textos y la justificación a utilizar y, como antes, se escribe a mano sobre el original. En las publicaciones pequeñas una misma persona suele desempeñar las funciones de redactor y corrector.

Una vez compuesto el artículo, se espera que ya casi haya adquirido su forma impresa final, aunque es probable que los teclistas o linotipistas hayan incurrido en errores al transcribir el texto original. En esta etapa, el artículo compuesto se denomina *galerada* o *prueba,* y el corrector la debe releer atentamente, marcar las erratas y devolverla al teclista, quien realizará las correcciones señaladas.

En la etapa de la primera prueba, otra persona se une al proceso de producción. El trabajo del maquetista consiste en tomar la prueba impresa del texto y diseñar el trazado de una página: el aspecto que tendrá la forma impresa final. Por supuesto, la mayoría de las publicaciones no contienen sólo texto, sino también fotografías y diagramas, junto con las correspondientes notas al pie. El maquetista ha de combinar equilibradamente las partes que constituirán el contenido final de la página a fin de dar a ésta una forma racional y atractiva. Con este fin se suele confeccionar una *maqueta* de la página y disponer en ella los diversos componentes como si se tratara de un rompecabezas de piezas irregulares. En esta etapa también se puede distribuir el texto de la prueba en la maqueta de la página, aunque algunos maquetistas se limitan a medir la longitud del texto con el fin de determinar el espacio que ocupará en la página terminada.

Durante este proceso sucede con frecuencia que el texto es demasiado largo o demasiado corto como para caber en el espacio que se le ha asignado. En este caso el redactor o el corrector tendrá que suprimir algunos párrafos o bien, si el texto fuera muy breve, añadir líneas para que alcance la longitud necesaria. Las supresiones o añadidos se incluyen en la prueba, de modo que, cuando el teclista la devuelva corregida, el artículo tenga la extensión requerida. Por último, se insertan en la página los diagramas, las fotografías y el texto de acuerdo a la maqueta confeccionada previamente, y entonces ya se puede mandar a imprimir la página.

A la luz de esta breve descripción del proceso editorial podemos apreciar que supone muchas etapas que significan una potencial pérdida de tiempo. En realidad el método que acabamos de reseñar constituye un híbrido en el cual, en ciertas fases, se emplean de forma aislada ordenadores independientes. Es probable que el artículo se escriba en un procesador de textos y que el mismo proceso tipográfico esté también informatizado gracias a una máquina no muy distinta a un procesador de textos normal, pero que incluye algunas instrucciones que posibilitan la selección de dife-

Un cambio revolucionario

Se escribe el artículo con un procesador de textos

Se imprime el artículo en forma de texto

Éste se corrige, prepara y envía a componer

El teclista produce una galerada

Maqueta preliminar de la galerada para comprobar la longitud del texto y el diseño

Se escribe el artículo en un procesador de textos

Se corrige y se prepara en la misma máquina o similar

La maqueta y el diseño de la página se efectúan en pantalla

rentes tipos, etc. A la vista de un sistema de esta clase, no haría falta un analista de sistemas profesional para sugerir posibles mejoras.

En primer lugar, imaginemos que en la oficina de la editora de la publicación hubiera un sistema de tratamiento de textos compatible con el que utiliza el autor. No sería necesario que éste produjera páginas impresas, sino tan sólo que entregara un disco o cinta con el artículo grabado magnéticamente, o incluso que transmitiera el artículo electrónicamente utilizando la red telefónica y un par de modems. La labor que cumplen el redactor y el corrector se podría realizar en pantalla. Aun cuando el proceso revirtiera, entonces, en esta etapa al método más tradicional, al imprimir la versión revisada del artículo en papel para enviársela a los teclistas, el sistema ya se habría modernizado un poco.

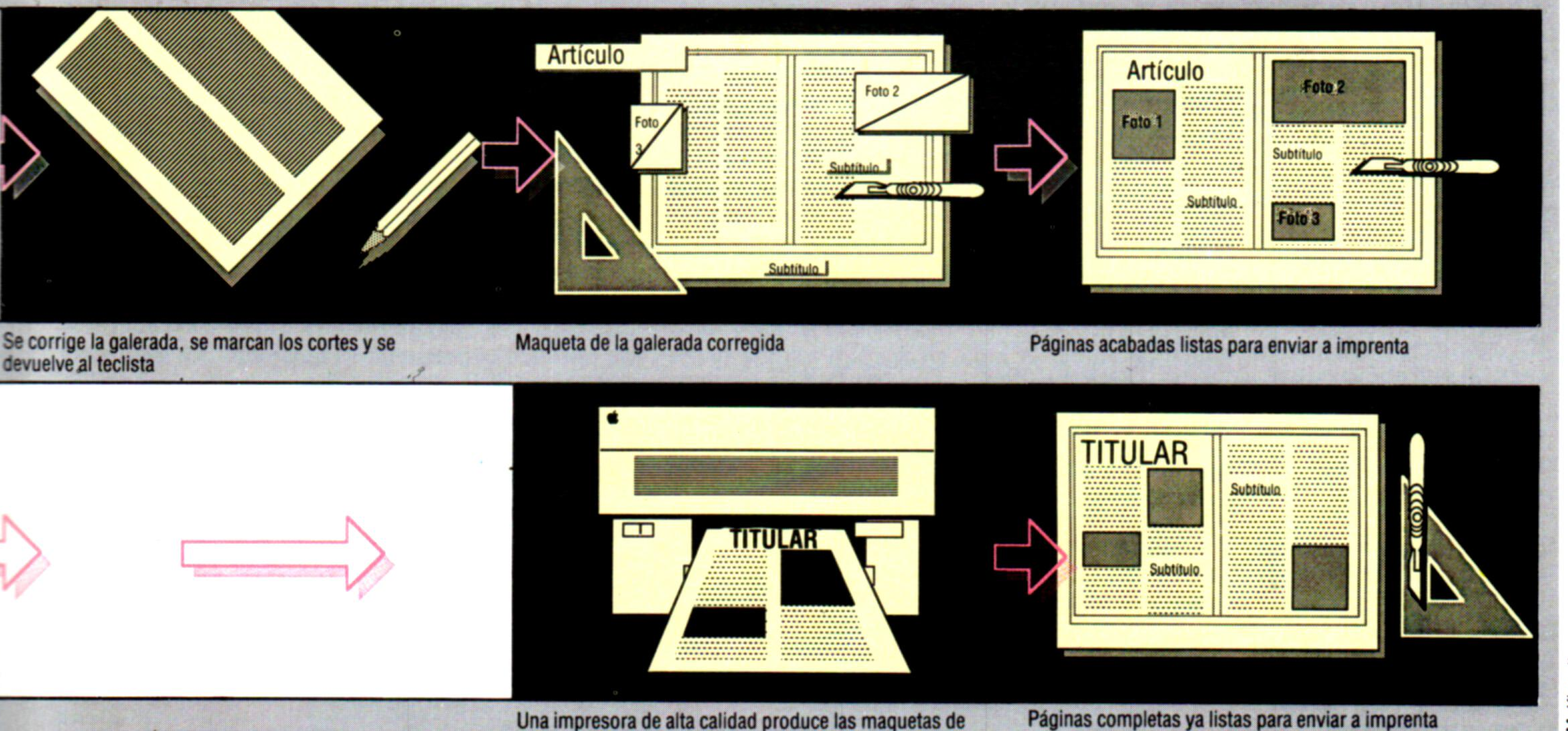
Avanzando otro paso, un sistema racionalizado aseguraría que la máquina tipográfica informatizada fuera capaz de leer el disco en el cual se almacenó la versión original revisada. De ser éste el caso, el artículo pasaría por la traducción del borrador original del autor a su forma tipográfica existiendo sólo como una serie de impresiones magnéticas en un disco hasta que se compusiera tipográficamente.

Se dice que este sistema es de «entrada directa»: el texto del artículo es digitado una sola vez, por su autor, y ulteriormente introducido en otras máquinas mediante carga desde disco o bien por transmisión electrónica entre secciones de un sistema informatizado homogéneo y de mayores dimensiones.

Sin embargo, las posibilidades de la nueva tecnología van mucho más allá de la manipulación de textos. Recientes desarrollos en entornos operativos de ordenador, en particular los sistemas WIMP (Windows, Icons, Mouse Program) han despejado el camino para que la informática intervenga en el trabajo del maquetista. Estos nuevos sistemas permiten que éste monte una página en pantalla, con frecuencia incluyendo fotografías digitalizadas y diagramas. Los diferentes tipos de letra y la disposición del texto se pueden manipular utilizando simplemente un ratón controlador. Luego la versión final de la página se puede imprimir directamente mediante una impresora de alta calidad.

A pesar de que en muchos sentidos la producción de periódicos es diferente de la producción de revistas, para ella rigen muchos de los mismos principios. De hecho, las necesidades de procesamiento rápido de la prensa diaria, sumado a su enorme volumen de ventas, significa que es probable que la nueva tecnología cause un mayor impacto en este campo de la industria editorial. La tecnología que se aplica actualmente en la prensa data, aproximadamente, de los años treinta y emplea procedimientos desarrollados a fines del s. XIX. No obstante, en la industria de la prensa se detectan indicios de cambios inminentes.

Muchos periódicos de provincias, en respuesta al desafío de los folletos publicitarios gratuitos de producción económica, ya han aplicado con todo éxito métodos de producción informatizada. Existen planes para que las empresas pequeñas y más flexibles desafíen al monopolio de los gigantes de la prensa produciendo un diario con alrededor de la quinta parte de personal que se requeriría normalmente en un periódico nacional, usando métodos de entrada directa, montaje en pantalla e impresión informatizada.



Se corrige la galerada, se marcan los cortes y se devuelve al teclista

Maqueta de la galerada corregida

Páginas acabadas listas para enviar a imprenta

Una impresora de alta calidad produce las maquetas de las páginas

Páginas completas ya listas para enviar a imprenta

Ian McKinnell

Un interesante efecto de la producción electrónica de un diario se puede observar en su distribución a través del país. Tradicionalmente, los periódicos nacionales se imprimen en un centro (o posiblemente dos) y luego los ejemplares impresos se distribuyen a través del país por tren y carretera. En el caso de un periódico producido por medios electrónicos, el método de distribución puede cambiar radicalmente. Aunque se puede producir el diario en un sitio, se puede imprimir en cinco o seis prensas de provincias transmitiendo el diario electrónicamente a través de enlaces telefónicos normales, reduciendo de modo significativo los costos de distribución. Además, resulta muy sencillo producir ediciones locales del diario nacional, añadiéndole páginas al contenido principal del periódico en los centros locales.

En la segunda parte de esta serie acerca de la informática en la industria editorial veremos cómo se están abriendo paso las nuevas técnicas de edición electrónica en el mercado de microordenadores, y mostraremos cómo se puede producir una revista, desde la creación del texto hasta el producto final, en un solo micro.

Página en blanco

Comenzamos un proyecto para diseñar un programa de hoja electrónica en BASIC

La mejor forma de pensar en un programa de una hoja de datos es como si se tratara de una hoja de papel electrónica, con una calculadora programable incorporada. El programa permite entrar números y otros datos en esta hoja en blanco de papel, y luego llevar a cabo cálculos específicos a partir de esta información. Para ello usted debe preparar una serie de fórmulas aritméticas o matemáticas en la hoja electrónica. En base a sus instrucciones, el programa de hoja electrónica utiliza estas fórmulas para efectuar cálculos con los datos presentes en la hoja. Luego coloca los datos nuevos (es decir, los resultados de los cálculos) en la hoja, en las posiciones que se especifican en las fórmulas.

De esto modo, si usted tiene un problema o una aplicación que le exijan trabajar con grandes cantidades de datos, la hoja electrónica es una herramienta ideal. Con ella puede añadir filas y columnas con sólo pulsar un botón, una configuración útil para totales de ventas, aplicación de fórmulas matemáticas a una serie de datos, cálculo de porcentajes para el IVA e impuesto sobre la renta, etc. En resumidas cuentas, con una hoja electrónica es posible realizar cualquier cosa que pueda hacer con lápiz, papel y una calculadora, sólo que lo puede llevar a cabo mucho más rápidamente y con mayor exactitud. Otra configuración útil de las buenas hojas electrónicas es que una vez que ha preparado una para que realice un conjunto dado de funciones, usted puede volver a utilizar esa hoja una y otra vez con distintos datos cada vez. Ésta es una facilidad sumamente útil si ha de efectuar con regularidad un conjunto de cálculos, como gastos de viaje semanales o comisiones sobre las ventas.

Por ejemplo, si usted lleva una pequeña empresa, puede reclamar el IVA que pagó sobre las compras en relación al IVA que tenga que indicar en su declaración de impuestos. Obviamente, antes de poder hacer esto, tendrá que calcular el IVA total que ha cargado en los artículos que ha vendido y restar el IVA total que ya ha pagado sobre los artículos que ha adquirido. Utilizando el método normal de lápiz y papel, tendría que calcular el IVA sobre cada artículo individual de forma manual y luego sumar todo el IVA al final, procedimiento que es tan tedioso como susceptible de error. Sin embargo, con una hoja electrónica, todo cuanto ha de hacer es prepararla para que calcule el IVA entrando la fórmula; supongamos, para multiplicar todos los artículos de una columna por 15 y dividir por 100. Luego se puede entrar el costo de las compras, y la hoja electrónica se encargará del resto.

Las hojas electrónicas se pueden utilizar para numerosos tipos de aplicaciones. Entre las domésticas se incluyen el presupuesto hogareño, los gastos del coche, análisis de préstamos y gastos generales. Entre los usos empresariales de las hojas electrónicas se incluyen facturación, libro mayor de ventas y compras, presupuestos, previsión y planificación,

Problema de impuestos

		A2*15/115			A5*15/100
		B2*15/115			B5*15/100
		C2*15/115			C5*15/100
		D2*15/115			D5*15/100

	1.COMPRADO	2.C. BRUTO	3.IVA	4.VENDIDO	5.PRECIO	6.IVA
A	MICRO	399.99	52.17	HOJA/ART. 1	120.00	18.00
B	IMPRESORA	249.99	32.60	PROG. CONV.	2000.00	300.00
C	PAPEL	20.00	2.60	HOJA/ART. 2	120.00	18.00
D	CINTA	7.24	0.94	DATOS CHISME	50.00	7.50
E	LÁPICES...	2.42	0.31			
F	ESCRITORIO	92.47	12.06			
G	UNID. DISCO	199.95	26.08			
H	LÁMPARA	24.87	3.24			
I	ELECTR.	57.32	7.47			
J	SANGRE	10.57	1.37			
K	SUDOR	1.00	0.13			
L	LÁGRIMAS	0.50	0.06			
M		IVA PAGADO	139.03		IVA CARGADO	343.50
N					IVA NETO	204.47

Caroline Clayton

Hoja de balance
El trazado de hoja electrónica que vemos aquí a modo de ejemplo muestra un típico problema de contabilidad: las cuentas de balance del IVA. El usuario ha de preparar las columnas 1, 2, 4 y 5 de la hoja electrónica, y ésta se debe programar para calcular las cifras de las columnas 3 y 6 y los totales del IVA. Por ejemplo, las cifras de la columna 3 se pueden hallar multiplicando la cifra correspondiente de la columna 2 por 15/115, y programarlas preparando las fórmulas en la celda de la columna 3 como se indica. Suele darse el caso, como aquí, de que se requieran fórmulas similares para una fila o columna completas de la hoja electrónica. Para que el usuario no tenga que prepararlas todas de forma manual, la mayoría de las hojas electrónicas incluyen una función para duplicar que permite repetir la fórmula de una celda a lo largo de una fila o una columna



análisis de costos, pérdidas y beneficios, costo y estimación de empleo, cálculos de comisiones sobre las ventas y modelado de proyectos, por nombrar sólo algunos. Los beneficios para el usuario son muchos e incluyen aumento de la productividad, exactitud y mejora de la presentación.

El programa de hoja electrónica que diseñaremos en esta serie le ofrecerá al programador de BASIC una esclarecedora visión de la forma como se escriben los programas comerciales de hoja electrónica y cómo funcionan. A medida que se vayan imprimiendo cada una de las partes del programa, se ofrecerá una explicación detallada acerca de cómo trabaja cada sección y lo que hace. El programa se ha escrito para Spectrum, Commodore 64, BBC Micro y Amstrad CPC 464/664; para cada sección del programa se imprimirá un listado en BASIC Spectrum y tipo Microsoft, junto con detalles para implementar el programa en otras máquinas. Se pretende que el programa sea ilustrativo; se lo podría utilizar como base para un programa mucho más sofisticado, a tenor del nivel de destreza que posea usted como programador y la cantidad de tiempo que desee dedicarle a la tarea.

Debido a la naturaleza compleja de las hojas electrónicas, nos hemos concentrado en reproducir las principales características de un programa de hoja electrónica, con el resultado de que en el programa la comprobación de errores existente es ínfima. Por consiguiente, habrá de tener gran cuidado al entrar sus datos y usar el programa, puesto que cualquier error hará que éste quede colgado. La ausencia de comprobación de errores obedece a dos motivos: el primero es la limitación de espacio y, el segundo, la velocidad de operación.

Puesto que el programa está escrito en BASIC, el tamaño de la hoja queda limitado a 15 columnas por 15 filas. Ello es para asegurar una velocidad de operación razonable y reducir la complejidad del programa. Observe que en todo momento sólo se puede visualizar una pequeña porción, o «ventana» de la hoja, debido a las limitaciones de las visualizaciones en pantalla. En las Commodore 64, BBC Micro y Amstrad, esta ventana es de cinco columnas por siete filas; en el Spectrum es de cuatro por seis.

Estructura celular

Cada uno de los 225 cuadraditos de nuestra hoja electrónica se denomina *celda*. Estas celdas están etiquetadas (en la primera fila) A1, A2, etc., hasta A15; B1, B2, etc. (en la segunda fila), hasta la decimoquinta fila (O1, O2, etc. hasta O15). Para desplazar el cursor por la hoja, usted simplemente usa las teclas del cursor de su máquina. La celda actual que esté señalando el cursor se visualiza en la esquina superior izquierda de la pantalla y se ilumina la propia celda en cuestión.

En cada celda usted puede bien entrar datos numéricos o bien establecer una nueva fórmula para manejar los datos. Éstos se visualizan en la pantalla en la celda adecuada y usted sólo cuenta con cinco dígitos por celda. Si en la celda actual hay una fórmula, la misma se visualiza en la línea de entrada de la parte inferior de la pantalla. Para entrar una fórmula nueva, simplemente seleccione la opción entrar fórmula y digite la nueva fórmula. El evaluador de fórmulas de este programa es bastante sofisticado, evaluando la fórmula de izquierda a derecha y con precedencia aritmética normal. Los operadores que puede manipular son +, −, *, /, ^ y paréntesis. Los operandos permitidos son nombres de celdas o números reales o enteros. Por lo tanto, si usted quisiera sumar los datos de las celdas A1, A2 y A3, y colocar luego la respuesta en la celda A4, entraría la fórmula A1 + A2 + A3 en la celda A4. Nuevamente, si quisiera calcular el IVA sobre el valor, supongamos, de B1, entonces colocaría en la celda B2 la fórmula B1* 0,15 y en la celda B3 la fórmula B1 + B2.

Funciones

Nuestro programa de hoja electrónica en BASIC incluye muchas características que la convierten en una poderosa herramienta para utilizar en el hogar. Éstas incluyen:

- Una pantalla de AYUDA, que imprime una lista de las opciones de que dispone el usuario.
- Una función ALMACENAMIENTO TEMPORAL que le permite guardar la hoja visualizada actualmente. Ésta es especialmente útil cuando quiere utilizar la hoja electrónica para probar valores diferentes.
- También puede RESTITUIR una hoja guardada temporalmente a la pantalla (es decir, después de haber probado otros valores, puede retomar otra vez sus datos originales).
- La función CALCULAR trabaja a través de la hoja de izquierda a derecha y de arriba abajo, calculando todas las fórmulas y alterando en consecuencia los contenidos de las celdas a medida que va avanzando. Por lo tanto, usted debe asegurarse de haber dispuesto los datos por un orden que no afecte a estos cálculos.
- LIMPIAR borra todos los datos de la hoja, de modo que ha de asegurarse, antes de utilizarla, de haber almacenado sus datos o bien haberlos guardado en cinta o disco.
- La función DUPLICAR le permite reproducir una fórmula en una gama de celdas, ahorrando en consecuencia mucho tiempo y mucho trabajo de digitación en caso de que usted tenga muchas fórmulas similares (como cálculos del IVA).
- Puede CARGAR y GUARDAR los datos en su hoja desde o en cinta.
- Asimismo, puede CARGAR y GUARDAR las fórmulas independientemente de los datos de la hoja. Esto es útil, porque le permite utilizar una hoja que se ha preparado para uso frecuente (como para calcular los gastos de viaje semanales) con datos nuevos y sin necesidad de preparar manualmente la hoja cada vez.
- La función TAB le permite llevar el cursor a cualquier posición de la hoja, sin tener que recorrer con él toda la pantalla.

Ofrecemos aquí los listados para preparar la pantalla y manipular los gráficos en el Commodore 64. En el próximo capítulo ofreceremos los listados correspondientes para BBC Micro, Spectrum y Amstrad. Ésta es la única sección del programa de hoja electrónica para la que ofreceremos listado separados. Esto se debe a que la manipulación de pantalla para cada máquina es totalmente diferente, mientras que el resto del programa es similar para cada ordenador (con la excepción de la versión para el Spectrum, que emplea distintos métodos de manipulación de series).

Rutinas gráficas

Aunque una buena parte del programa de hoja electrónica se puede ofrecer en forma de listado común, los métodos en que los cuatro micros manipulan la visualización de información en la pantalla son distintos. Para producir visualizaciones atractivas y manipulación en pantalla en cada máquina, proporcionamos listados separados de las rutinas para gráficos para cada ordenador. Comenzamos ofreciendo los listados para el C64. Estos listados manipulan la impresión de la cuadrícula de la hoja electrónica, la impresión de datos en las celdas de la hoja electrónica y el movimiento del cursor en la hoja. Tras digitar la sección de programa adecuada y ejecutarla, verá en la pantalla la cuadrícula (junto con un conjunto inicial de datos) y podrá desplazar el cursor por la hoja. No obstante, la mayoría de las funciones de la hoja electrónica que mencionamos no estarían disponibles todavía, porque serán el tema que desarrollaremos en futuros capítulos



Commodore 64

Liz Heaney

```
10 REM **** HOJA ELECTRONICA CBM 64 ****
15 REM ** PREPARAR SERIES CARACTERES CBM **
20 FOR I=1 TO 7:L7$=L7$+CHR$(195):NEXT
30 L5$=LEFT$(L7$,5):T$=CHR$(178):B$=CHR$(194)
40 S5$="     ":X$=CHR$(219):I$=CHR$(177)

100 GOSUB 3000:REM PREPARAR MATRICES Y VARIABLES
110 GOSUB 1000:REM IMPRIMIR PANTALLA
120 GOSUB 1700:REM IMPRIMIR DATOS EN PANTALLA
130 GOSUB 1100:REM RUTINA PRINCIPAL EXPLORACION DEL
    TECLADO
999 STOP
1000 PRINT CHR$(147);CHR$(145);CHR$(5):POKE
    53280,6:POKE 53281,6
1005 PRINT"           C O L U M N A S"

1006 PRINT
1007 PRINT "FILA      1.      2.      3.      4.      5."
1010 PRINT "        "CHR$(176);L5$;T$;
    L5$;T$;L5$;T$;L5$;T$;L5$;CHR$(174)
1020 FOR C=1 TO 7
1030 PRINT "  "CHR$(C+64);".          "B$;S5$;B$;S5$;
    B$;S5$;B$;S5$;B$;S5$;B$
1040 PRINT "  ";L7$;X$;L5$;X$;L5$;X$;L5$;X$;
    L5$;X$;L5$;CHR$(179)
1050 NEXT C
1060 PRINT "  "CHR$(C+64);".          "B$;S5$;
    B$;S5$;B$;S5$;B$;S5$;B$;S5$;B$
1070 PRINT "  ";L7$;I$;L5$;I$;L5$;I$;L5$;
    I$;L5$;I$;L5$;CHR$(189)
1080 RETURN
1100 P$=CHR$(Y+64)+MID$(STR$(X),2,2):PRINT
    CHR$(19);"CELDA:";P$;"  "
1110 GET A$:IF A$="" THEN 1110
1120 IF A$=CHR$(29) THEN 1200:REM MOVER DERECHA
1130 IF A$=CHR$(157) THEN 1300:REM MOVER IZQUIERZA
1140 IF A$=CHR$(17) THEN 1400:REM MOVER ABAJO
1150 IF A$=CHR$(145) THEN 1500:REM MOVER ARRIBA
1152 IF A$=CHR$(133) THEN GOSUB 6000:REM F1 IMPRIMIR
    PANTALLA AYUDA
1155 IF A$=CHR$(137) THEN GOSUB 2000:REM F2 ENTRAR
    FORMULA
1158 IF A$=CHR$(134) THEN GOSUB 5150:REM F3 ALMACENAR
    HOJA ACTUAL
1160 IF A$=CHR$(135) THEN GOSUB 2300:REM F5 CALCULAR
    HOJA
1165 IF A$=CHR$(13) THEN RETURN
1170 IF A$>"0" AND A$<"9" THEN GOSUB 2100:REM ENTRAR
    DATOS NUMERICOS
1180 IF A$=CHR$(139) THEN GOSUB 5000:REM F6 BORRAR
    HOJA
1185 IF A$=CHR$(138) THEN GOSUB 5100:REM F4 RETOMAR
    HOJA ANTERIOR
1187 IF A$="G" THEN GOSUB 5200
1188 IF A$=CHR$(136) THEN GOSUB 5700:REM F7 DECIDIR COL
    O FILA
1189 IF A$=CHR$(140) THEN GOSUB 7000:REM F8 RUTINAS
    CARGAR/GUARDAR
1190 GOTO 1110:REM VUELTA A EMPEZAR
1200 REM **** MOVER DERECHA ****
1210 IF X=15 THEN 1100
1220 IF X=H2 THEN GOSUB 1600:X=X+1:H1=H1
    +1:H2=H2+1:GOTO 1270
1230 GOSUB 1660:LET X=X+1:GOSUB 1650:GOTO 1100
1270 GOSUB 1800:GOSUB 1700:GOTO 1100
1300 REM **** MOVER IZQUIERDA ****
1310 IF X=1 THEN 1100
1320 IF X=H1 THEN GOSUB 1600:X=X-1:H1=H1-1:
    H2=H2-1:GOTO 1370
1330 GOSUB 1600:LET X=X-1:GOSUB 1650:GOTO 1100
1370 GOSUB 1800:GOSUB 1700:GOTO 1100
1400 REM **** MOVER ABAJO ****
1410 IF Y=15 THEN 1100
1420 IF Y=V2 THEN GOSUB 1600:Y=Y+1:V1=V1
    +1:V2=V2+1:GOTO 1470
1430 GOSUB 1600:LET Y=Y+1:GOSUB 1650:GOTO 1100
1470 GOSUB 1850:GOSUB 1700:GOTO 1100
1500 REM **** MOVER ARRIBA ****
1510 IF Y=1 THEN 1100
1520 IF Y=V1 THEN GOSUB 1600:Y=Y-1:V1=V1
    -1:V2=V2-1:GOTO 1570
1530 GOSUB 1600:LET Y=Y-1:GOSUB 1650:GOTO 1100
1570 GOSUB 1850:GOSUB 1700:GOTO 1100
1600 REM **** APAGAR CURSOR ****
1610 CU=1023+40*(V(Y+1-V1)+H(X+1-H1):
    POKE CU,PEEK(CU)-128
1620 POKE CU+1,PEEK(CU+1)-128:POKE CU+2,
    PEEK(CU+2)-128
1630 POKE CU+3,PEEK(CU+3)-128:POKE CU+4,
    PEEK(CU+4)-128:RETURN
1650 REM **** ENCENDER CURSOR ****
1660 CU=1023+40*(V(Y+1-V1)-1)+H(X+1-H1):
    POKE CU,PEEK(CU)+128
1670 POKE CU+1,PEEK(CU+1)+128:POKE CU+2,
    PEEK(CU+2)+128
1680 POKE CU+3,PEEK(CU+3)+128:POKE CU+4,
    PEEK(CU+4)+128
1690 GOSUB 1900:RETURN
1700 REM **** IMPRIMIR DATOS EN HOJA ****
1710 FOR I=0 TO 7
1720 PRINT CHR$(19);:FOR C=1 TO V(I+1)-1:
    PRINT CHR$(17);:NEXT C
1730 FOR J=0 TO 4
1735 P$=MID$(STR$(MAT(I+V1,J+H1)),2)
1740 PRINT TAB(H(J+1)-1);"          ";CHR$(145)
1745 PRINT TAB(H(J+1)+4-LEN(P$));P$;
1750 NEXT J,I
1760 GOSUB 1650:RETURN
1800 REM **** IMPRIMIR NUMERO COLUMNA ****
1810 PRINT CHR$(19);CHR$(17);CHR$(17);CHR$(17);
1820 FOR I=H1 TO H2:PRINT TAB(7+6*(I-H1))
    I;CHR$(157);".  ";
1830 NEXT I:RETURN
1850 REM **** IMPRIMIR NUMEROS FILA ****
1860 PRINT CHR$(19);:FOR I=1 TO 5:PRINT CHR$(17);: NEXT
1870 FOR C=V1 TO V2
1880 PRINT TAB(1)CHR$(C+64);CHR$(17)
1890 NEXT C:RETURN
1900 REM *** FORMULA DE CELDA ACTUAL ***
1920 LET D$=F$((Y-1)*15+X)
1930 GOSUB 1950:REM MOVER CURSOR HASTA LINEA
    ENTRADA
1935 PRINT CHR$(18);"FORMULA:
1940 PRINT CHR$(145);CHR$(18)"FORMULA:";D$
1945 PRINT CHR$(19):RETURN
1950 REM **** CURSOR A LINEA ENTRADA ****
1960 PRINT CHR$(19);:FOR K=1 TO 22:PRINT
    CHR$(17);:NEXT K
1970 RETURN

3000 REM **** PREPARAR MATRICES ****
3010 DIM H(5),V(8),ST(20),ST$(20),E$(20),G$(20)
3020 FOR C=0 TO 4
3030 H(C+1)=6*C+10:REM CALC POS X
3040 NEXT C
3050 FOR C=1 TO 8
3060 V(C)=2*C+4:REM CALC POS Y
3070 NEXT C
3075 X=1:Y=1
3080 H1=X:H2=X+4:V1=Y:V2=Y+7
3090 REM ***** MATRICES HOJA *****
3100 DIM MAT(15,15):DIM MC(15,15)
3110 FOR I=1 TO 15:FOR J=1 TO 15
3120 MAT(I,J)=I*J
3130 NEXT J,I
3140 DIM F$(225)
3150 RETURN
```



Jugar con palabras

Finalmente veremos cómo el FORTH manipula las series

Ya hemos visto la palabra .", que es bastante fácil de usar, pero hay otras manipulaciones de series que son más complicadas. Existen dos problemas principales: en qué zona de la memoria almacenar la serie y cómo hacerlo. Tales consideraciones son engañosas, por lo que, como es natural, querrá definir palabras potentes que en el momento de utilizarlas le permitan olvidar todos los detalles.

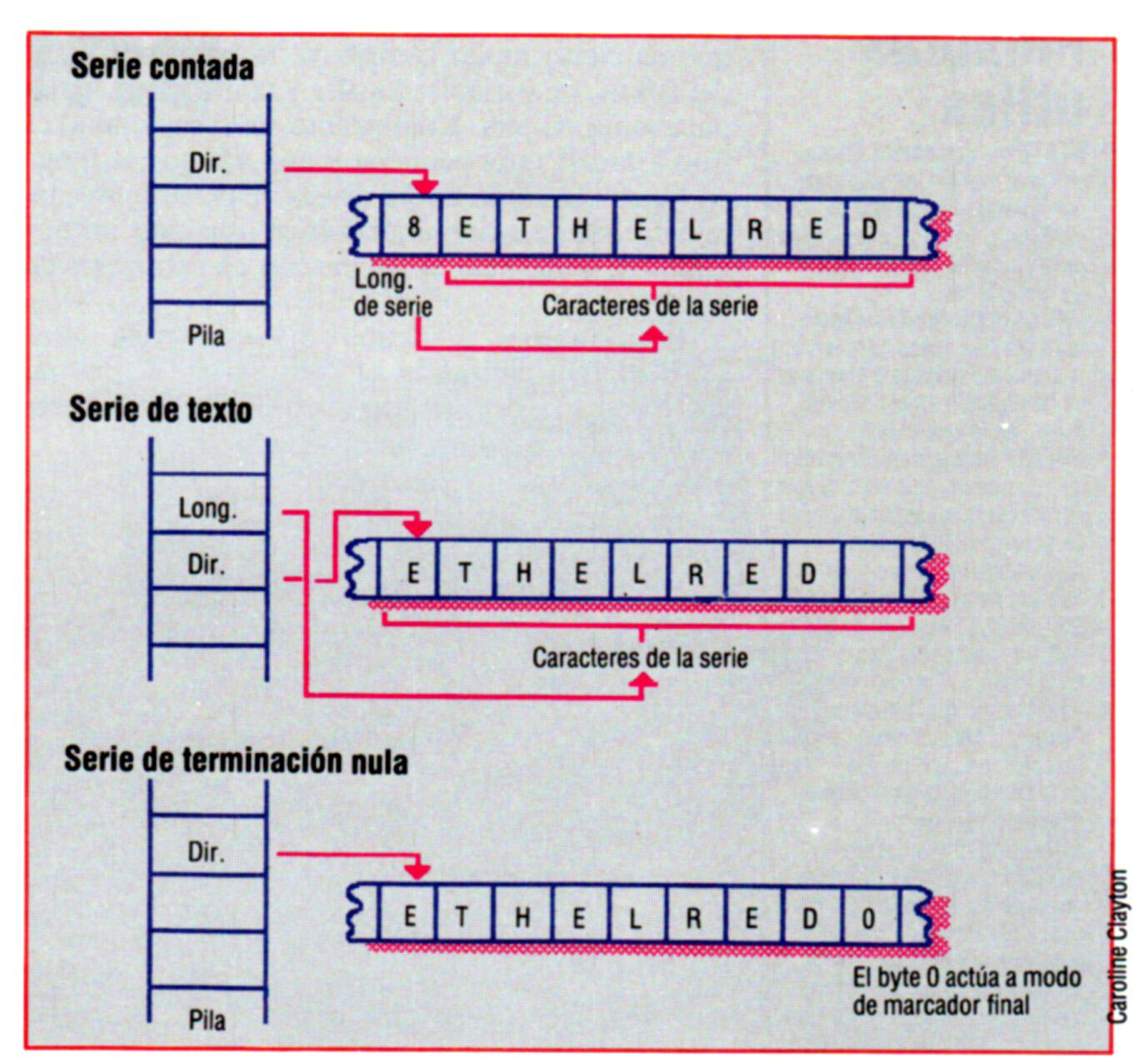
En primer lugar, hay dos lugares principales donde almacenar una serie: en los campos de parámetros asignados (ALLOT) de palabras del diccionario, y en una zona llamada *relleno*. El relleno es para el almacenamiento de series muy temporal, porque el sistema FORTH lo emplea con mucha frecuencia. Por ejemplo, si se incluye algo en el diccionario, o si se interpreta una palabra FORTH desde el teclado, el relleno puede quedar desplazado o bien sobreescrito. En segundo lugar, en FORTH hay dos maneras de almacenar una serie. En ambos casos los caracteres ASCII de la serie se almacenan por orden en algún lugar de la memoria; las diferencias radican en cómo especifique la longitud de la serie.

En una *serie contada,* la longitud (menos de 225 bytes) se almacena como un único byte inmediatamente antes de los caracteres. Para aludir a una serie de este tipo, puede utilizar un número, que es la dirección de este byte de cuenta. Por lo tanto, aunque no puede colocar la serie propiamente dicha en la pila, tras haber establecido la serie puede colocar su dirección en la pila. En el caso de una *serie de texto,* la longitud no se almacena para nada como parte de la serie. Esto significa que para aludir a la serie necesita dos números, la dirección del primer carácter, así como la longitud.

Las series contadas son más fáciles de manipular en la pila (el número adicional necesario para las series de texto hace que las manipulaciones de pila se vuelvan desproporcionadamente complicadas). Sin embargo, tienen un límite de 255 bytes y no se pueden utilizar para subseries, series que constituyen partes de series más grandes, porque en medio de la serie completa, tendría que insertar un byte de cuenta para la subserie.

Veamos algunas palabras que definen variables de serie. Una variable de serie mantiene su texto en su campo de parámetros, de modo que una vez que adjudica (ALLOT) una cierta cantidad de espacio para una serie, el mismo permanece fijo, por lo que nunca podrá destinarle series más largas. $VARIABLE, por consiguiente, requiere que especifique no sólo un valor inicial, sino también una longitud máxima para ulteriores asignaciones. La longitud máxima habrá de ser mayor que el valor inicial.

Una palabra útil es ASCII. La mayoría de los FORTH tienen incorporada esta palabra, si bien esto no es preceptivo en el estándar. Apila el código ASCII del carácter que le sigue y, si se utiliza en una definición de dos puntos, compila un literal numérico para este número de modo que se lo apila en el tiempo de ejecución.



Formatos de series
El FORTH permite retener series en la memoria de tres formas diferentes. Cada método emplea un número de dirección (que se puede manipular fácilmente mediante la pila del FORTH) que señala la zona de memoria que contiene a la serie. Sin embargo, las tres técnicas utilizan distintos métodos para determinar la longitud de la serie. El formato «serie contada» almacena esta información en el primer byte del área de memoria que contiene la serie, mientras que el formato «serie de texto» retiene este número por separado en la pila, permitiendo la manipulación de pila del mismo. Algunas versiones de FORTH soportan el formato «terminación nula», en el cual el fin de la serie se indica simplemente mediante un byte cero

```
32 CONSTANT BL (código ASCII para un espacio)
:ASCII(– –código ASCII) (utilizado para formar
        carácter ASCII)
   BL WORD (tomar siguiente palabra)
   1+C@ (y tomar el código ASCII para
        su primer carácter)
   STATE @ IF (si compilando definición dos puntos)
      [COMPILE] LITERAL(LITERAL es inmediata)
   THEN
;IMMEDIATE(ASCII tiene acciones de compilaciones
especiales)
;$ VARIABLE(tamaño– –)(utilizado en la forma —)
      (tamaño $VARIABLE nombre inicializador")
      (se prepara el campo de parámetros para conte-
      ner 1 byte para la)
      (longitud máxima y luego una serie contada
      para su valor)
   CREATE
   255 MIN (asegura que el tamaño no sea
   demasiado grande)
   1 ALLOT (para longitud máx.)
   ASCII "WORD(tamaño, inicializador como serie
   contada)
   DUP C@ ROT MAX(inicializador, longitud máx.)
   DUP HERE 1— C!(rellenar longitud máx.)
   HERE SWAP 1+ALLOT (inicializador, dirección
   para colocarlo)
   OVER C@1 + CMOVE (copiar inicializador)
DOES>(pfa– –pfa+1)(apila el valor como
       serie contada)
   1+
;
```

Palabras útiles

STATE (—dirección) Es una variable que lee las palabras inmediatas. Si su valor es verdadero (no 0), se las utiliza para compilar una definición de dos puntos.
WORD (delimitador—serie contada) Se utiliza para leerle la siguiente palabra a lo que ya se ha digitado en el teclado. P. ej., es lo que utiliza CREATE para tomar el nombre de sus nuevas palabras. Suele ser 32 (para un espacio), pero se pueden utilizar otros delimitadores, como " en nuestra definición de $VARIABLE. Algunas palabras del FORTH estándar hacen este truco, como ." and (usando "and) como delimitadores. Tenga cuidado al emplear la palabra con rapidez. Es probable que se almacene en el relleno, donde no permanecerá mucho tiempo.
EXPECT (dirección, número máximo de caracteres—) Se utiliza para tomar nuevas entradas del teclado. Los caracteres, hasta el número máximo especificado, se leen en la memoria comenzando por la dirección dada. Interrumpe la lectura cuando se pulsa ENTER o al digitar la cantidad máxima.
SPAN (—dirección) Esta variable la establece EXPECT para mostrar cuántos caracteres se han leído. Sólo la posee el FORTH-83; los más antiguos ponen caracteres NUL tras la serie para señalar dónde termina.
PAD (—dir. del relleno)
CMOVE (desde dirección, hasta dirección, número de caracteres— —) Copia el número de bytes especificado, comenzando desde la dirección «desde», en la memoria que empieza en la dirección «hasta», de modo que se utiliza para copiar series. Atención: si la zona de memoria «hasta» está después de la zona «desde», y las dos zonas se superponen, entonces algunos de los bytes de la zona «desde» se sobreescribirán antes de que se los lea, por lo que no será una copia fidedigna.
CMOVE> (desde dirección, hasta dirección, número de caracteres— —) Esta palabra, existente sólo en FORTH-83, evita la trampa de CMOVE al copiar los bytes por un orden diferente. No obstante, tiene una trampa propia cuando la zona «hasta» está antes de la zona «desde».
COUNT (serie contada — — dirección, longitud de serie texto) La serie queda igual, pero COUNT realiza una conversión entre las dos formas distintas de referirse a ella en la pila

Ésta es una definición bastante complicada, a lo que en cierto modo contribuye la falta, por parte del FORTH, de variables locales y también sus manipulaciones de pila. No obstante, su complejidad es sólo a escala muy pequeña y una vez que la tenga en funcionamiento podrá olvidarse de su funcionamiento interno, permitiéndole pensar más en términos de series que de secuencias de caracteres en la memoria.

Ahora vamos a definir $@ y$! para estas $VARIABLES, utilizando el formato de series de texto (dirección y longitud) en la pila para hacer referencia a estos valores.

```
:S@ (pfa+ 1 para variable de serie– –serie de
texto)
    COUNT
;
:$! (serie de textos, pfa+1 para variable de
    serie– –)
    (trunca el valor si es demasiado grande para
    la variable)
DUP 1–C@ROT       (dir,pfa, long. máx.,
                  long. real)
MIN               (dir,pfa+1, long. a usar)
OVER OVER SWAP    (dir,pfa+1,long.,
                  long.,pfa+1)
C!                (almacenar nueva long.)
SWAP 1+SWAP       (dir,pfa+2,long.)
CMOVE
;
```

Ya puede establecer variables en serie, como en:

```
0 $VARIABLE DIGITOS 0123456789"
```

(El 0 se incrementa automáticamente a 10 para hacer frente a los 10 caracteres presentes en realidad. Pero después de eso, la longitud máxima de la variable en serie queda fija en 10.)

```
DIGITOS $@TYPE
```

(visualiza 0123456789)

```
255 $VARIABLEXPAD XXX"
```

(un relleno alternativo, más estable, con espacio para 255 caracteres.)

Sería agradable poder cambiar los valores de las series desde el teclado. He aquí dos palabras, " y $INPUT, que le permiten realizar el equivalente de las sentencias de asignación del BASIC LET A$="..."

```
:"        (–.–dirección, longitud de serie texto)
          (utilizada fuera de definiciones de dos puntos
          en forma — "texto")
ASCII"WORD COUNT     (tomar serie de texto)
XPAD $!              (colocarla en XPAD)
XPAD $@              (dejar dir. y long.)
;
```

La palabra " toma su serie de la misma manera en que palabras como VARIABLE toman el nuevo nombre, motivo por el cual no funcionarán adecuadamente en definiciones de dos puntos. Sin embargo, puede utilizarla como en:

```
20 $VARIABLE MASCOTA gato"
"cotorra"MASCOTA $!MASCOTA $@TYPE
"pangolín"MASCOTA $! MASCOTA $@TYPE
```

He aquí una palabra para imitar sentencias de entrada:

```
:$INPUT (pfa+1 de variable en serie– –)
        (espera que digite una serie y la copia en la
        variable.)
  DUP DUP 1+        (pfa+1,pfa+1,pfa+2)
  SWAP 1–C@2–       (pfa+1,pfa+2,long.
                    máx.–2)
  EXPECT
  SPAN@ SWAP C!
;
```

El programa informa a EXPECT que no tome más caracteres para los que haya sitio en la variable. No obstante, una versión de FORTH permite escribir después de la serie un carácter NUL (nulo) adicional. De modo que, por razones de seguridad, se le ha restado 2 a la longitud máxima. De lo contrario, el NUL adicional podría alterar el encabezador de la siguiente palabra del diccionario.

En FORTH-79 o figFORTH, EXPECT escribirá un NUL o dos en todo caso, que en realidad son vitales. Los FORTH más antiguos no poseen SPAN y, en consecuencia, tendrá que escribir su propia palabra para hallar la longitud de la serie, contando los caracteres hasta el NUL.

Ahora, por fin, podemos definir una versión en FORTH del programa en BASIC más satisfactorio:

```
10 PRINT "Cómo te llamas?"
20 INPUT A$
30 PRINT "Encantado de conocerte, ";A$;"."
40 PRINT "Cuídate! Adiós."
```

Un programa comparable en FORTH, que emplea instrucciones definidas por nosotros, es:

```
30 $VARIABLE NOMBRE

:HOLA!
  ."Hola! Soy el ordenador. Quién eres tú?"CR
  NOMBRE $INPUT
  ."Vaya! Es un nombre increíble, chico,"NOMBRE
  $@TYPE
  ."."(digitar punto aparte)CR
  ."Cuídate, oyes? Que lo pases bien."
  CR
;
```

El gran mérito del BASIC es que es muy fácil escribir programas cortos, a causa de sus útiles facilidades para cosas como aritmética de punto flotante, matrices y series. Donde el BASIC pierde soltura es cuando se trata de escribir programas más largos. Usted nunca puede abstraerse de la estructura de bajo nivel del programa: los números de línea, por ejemplo. Por el contrario, observe nuestra palabra HOLA! en FORTH. Comprenderla no le supondrá mucho esfuerzo, sabiendo aproximadamente lo que hacen $INPUT, $@ y otras palabras. Si quiere estudiarla con mayor profundidad, vuelva a las definiciones de estas palabras, pero así y todo sólo necesitará saber *qué* hacen (lo que esperan que haya en la pila y lo que dejan en ella) e ignorar todo acerca de *cómo* lo hacen. Por consiguiente, en FORTH no es necesario tener presente la estructura de bajo nivel.

La conclusión de todo esto es: aunque el FORTH parte como un lenguaje menos poderoso que otros, usted puede valerse de su ampliabilidad para restituir el equilibrio. Y, lo que es más, puede continuar ampliándolo hasta que llegue a superar con mucho ese otro lenguaje.



Gran Hermano

El recién aparecido PC/AT, de IBM, que se aproxima a un miniordenador en cuanto a potencia y capacidad, establece un nuevo estándar para micros

A causa del prestigio alcanzado por IBM, en 1982 los microordenadores se pusieron repentinamente de moda en el ámbito empresarial, y el PC acaparó un considerable sector del mercado. Esto tuvo consecuencias cuyo impacto cabal ni la propia IBM habría podido prever. Gran parte de la tecnología utilizada en el IBM PC original se compró a otros proveedores, como el procesador Intel 8088, las unidades de disco de 5 ¼ pulgadas y, lo más importante, el sistema operativo PC-DOS. En su momento, ninguno de estos componentes representaba una tecnología avanzada, lo que suscitó muchas críticas, no obstante lo cual se vendió un gran número de máquinas, lo que, como es natural, generó una enorme base de software. Otros fabricantes, sintiéndose desplazados del mercado, siguieron el ejemplo de IBM y compraron una tecnología idéntica para producir «clonos IBM»: ordenadores que ejecutaban software IBM. Esto tuvo como consecuencia que el IBM PC se constituyera en el estándar industrial *de facto*.

Desde entonces, la reducción del precio de los chips de memoria, junto con unos procesadores cada vez más potentes, ha dado como resultado que las máquinas de gestión se equipen con 256 K como estándar (el PC original tenía sólo 64 K de RAM), lo que a su vez ha llevado a las firmas de software a producir programas más complejos y más ávidos de memoria.

La cima actual de esta tendencia la representa el PC/AT (*Personal Computer/Advanced Technology:* ordenador personal de tecnología avanzada). Este impresionante ordenador, que comienza a aproximarse a la potencia y capacidad de un miniordenador, está equipado con uno de los últimos procesadores: el Intel 80286. Este chip está equipado con un bus de datos de 16 bits y un bus de direcciones de 24 bits similar al del chip 8086 utilizado tanto en el Olivetti M-24 como en la gama Apricot, permitiendo que la máquina direccione más de 16 Mbytes de memoria. No obstante, el chip posee asimismo la capacidad para *gestión de memoria virtual,* lo que significa que la capacidad de memoria real que puede utilizar es de más de un gigabyte (1 000 Mbytes).

La gestión de memoria virtual es un proceso en virtud del cual el procesador puede tratar a su RAM disponible y el almacenamiento de apoyo como «memoria principal». Al ejecutar un programa, puede ser que en la RAM disponible no quepan todos los datos y el programa, de modo que el resto se retendrá en discos de alta velocidad. Cuando el ordenador requiere una información determinada y ve que la misma no está en RAM, los datos correspondientes se leen del disco en la zona de memoria sin utilizar, donde se actuará sobre ellos de la forma pertinente. De este modo, el ordenador *parece* tener muchísima más memoria de la que en realidad posee.

Chris Stevens



Tecnología avanzada
La entrada de IBM en el mercado del ordenador personal determinó un cambio en la actitud comercial hacia los microordenadores. Actualmente, la nueva máquina de la empresa, el IBM PC/AT, es uno de los micros más potentes que existen para uso de gestión. Operando hasta tres veces más rápido que el IBM PC, el AT también es capaz de acceder a hasta tres megabytes de memoria

Esta técnica se ha adaptado para aplicarla en el PC/AT. El objetivo original de la gestión de memoria virtual era el de usar la RAM en conjunción con métodos de almacenamiento más económicos, como discos. Sin embargo, con la caída de precios de los chips de RAM, esto no es fuente de preocupación en el sistema PC/AT. El problema está en el sistema operativo.

El PC-DOS, y los MS-DOS estrechamente relacionados con el mismo, son capaces de direccionar 640 K de memoria. Cuando se desarrolló el PC-DOS, esta memoria se consideraba más que suficiente para cualquier aplicación previsible. No obstante, muchos modernos paquetes de software integrado se están aproximando rápidamente al límite de las capacidades del OS. Por consiguiente, se había de encontrar algún método para añadir memoria dentro de las limitaciones impuestas por el PC-DOS.

La respuesta ha sido tratar la memoria restante como *discos de silicio*. Éstos son bancos de memoria que pueden ser de cualquier tamaño (aunque en el PC/AT el tamaño por defecto es de 64 K) hasta igualar el de la memoria direccionable, y que el procesador trata como si fuera información retenida en una unidad de disco. Esto significa que el procesador no puede procesar la memoria directamente, sino que debe ir a buscar la información en bloques, que se puedan trasladar a la memoria direccionable y manipular allí. Cuando ya no se nece-

IBM PC/AT

DIMENSIONES	540 × 420 × 160 mm
CPU	Intel 80286 trabajando a 6 MHz
MEMORIA	Base Model, 256 Kbytes; Enhanced Model, 512 Kbytes ampliables a 3 Mbytes
PANTALLA	Visualización de textos de 80 × 25 caracteres, con una resolución máxima de 640 × 200 pixels
INTERFACES	Ocho ranuras de ampliación para una amplia gama de interfaces
LENGUAJES DISPONIBLES	Todos los lenguajes principales utilizados comúnmente
DOCUMENTACION	Como siempre, tratándose de IBM, la documentación es exhaustiva, si bien algo formal
VENTAJAS	El PC/AT es uno de los microordenadores disponibles a nivel comercial más potentes del mercado, construido por el mayor fabricante de ordenadores del mundo. Es difícil ver en qué podría fallar
DESVENTAJAS	Incompatibilidad con parte del software PC existente

sitan más los bloques, la información actualizada o el programa se pueden volver a «escribir» en el disco de silicio. Por supuesto, aunque se traten como discos, el hecho de que los discos de silicio en realidad sean chips de RAM significa que el acceso es mucho más rápido que el de una unidad de disco convencional.

El PC/AT viene en dos configuraciones básicas. El sistema de precio mínimo, llamado Base Model 1 (modelo básico 1), contiene 512 K de RAM y una sola unidad de disco, mientras que el Enhanced Model (modelo mejorado) posee una unidad de disco rígido adicional de 20 Mbytes y un adaptador serie/paralelo.

Unidades de disco

Las unidades de disco incorporadas en el PC/AT, aun siendo discos flexibles de 5 ¼ pulgadas, representan una sustancial mejora respecto a las incorporadas en las versiones originales del IBM PC, que eran algo lentas y ruidosas, con una capacidad de apenas 160 K; las instaladas en el PC/AT tienen una capacidad de 1,2 Mbytes. Esto se debe en parte a la tecnología mejorada del diseño de las unidades y también a las mejoras introducidas en el DOS. El PC-DOS empaquetado en la máquina es la versión 3, que lee y escribe 15 sectores por pista en lugar de los ocho del PC. No obstante, para conservar la compatibilidad con versiones anteriores del IBM PC, la nueva máquina puede detectar discos de formato PC y adaptarse por sí misma a ellos; los PC, sin embargo, no pueden leer discos del PC/AT.

En el PC-DOS 3, además de las mejoras introducidas para sacar partido de la mayor capacidad de las unidades de disco, se han incorporado algunas instrucciones adicionales para mejorar su propio rendimiento. ATTRIB designa un archivo como de lectura solamente y LABEL permite asignar a un disco una «etiqueta de volumen». SELECT y COUNTRY son instrucciones similares, permitiendo la primera de ellas elegir el trazado del teclado y el formato hora/fecha requerido, y la segunda pasando automáticamente la hora/fecha y trazado del teclado a los de un determinado país. SHARE permite compartir archivos, FCBS especifica el número de bloques de control de archivos que se pueden abrir en un momento dado, y DEVICE permite instalar en el ordenador el tamaño y número de «discos virtuales». Por último, LASTDRIVE permite establecer la cantidad de unidades que puede utilizar el sistema.

En relación al espinoso problema de la compatibilidad, quizá sea sorprendente descubrir que el PC/AT no es totalmente compatible con el PC original. Esto se debe en parte a que las unidades de disco utilizadas en la nueva máquina, aunque mejoradas, son incapaces de hacer frente a algunas de las sofisticadas técnicas que se han desarrollado para impedir copiar el software PC. Otra dificultad reside en que algunos de los chips del PC/AT se han modificado respecto al original y, aunque se supone que son compatibles, el software que interroga directamente a estos chips podría encontrarse con que algunas de las direcciones han cambiado. La más notoria de ellas es la BIOS ROM, núcleo de la compatibilidad con el IBM PC, en la que se han introducido algunas mejoras. Lamentablemente, esto significa que también se han efectuado modificaciones en las direcciones de entrada de ciertas rutinas.

Este problema de incompatibilidad alcanza también al procesador. Aunque al Intel 80286 lo fabrica la misma empresa que produjo el procesador 8088 del PC, algunas de las instrucciones son distintas: el software para una máquina que use las instrucciones incompatibles no se ejecutará en la otra. Por último, el IBM PC original contenía interruptores DIP que en el PC/AT se han sustituido por CMOS RAM.

Debido a esto, se han tenido que ajustar muchos productos de software para encajar en el nuevo formato, principalmente con respecto a sus códigos de protección. Quizá el caso más notable sea el del simulador Microsoft Flight (FS1), que se considera como la prueba de compatibilidad IBM por antonomasia. Por lo tanto, Microsoft ha introducido el FS2, un paquete mejorado.

A pesar de estas dificultades, el IBM PC/AT es una máquina excelente, y casi todas las críticas que suscitara el PC original han tenido una respuesta positiva. Entre las mejoras destaca el teclado que, desde el punto de vista del tacto, es sin lugar a dudas el mejor que se haya producido jamás para un micro.

Como cabría esperar, la nueva máquina es considerablemente más rápida que el IBM PC original, velocidad que en algunas aplicaciones llega a ser tres veces superior. Esto, unido a la vasta memoria a la que puede acceder esta máquina, arroja muy pocas dudas acerca del futuro del PC/AT.

Apariencia engañosa

La técnica de *gestión de memoria virtual* se desarrolló originalmente para usar en ordenadores centrales, y se utiliza para darle a un ordenador la apariencia de tener mucha más memoria de la que posee en realidad. El principio de la gestión de memoria virtual es bastante simple. A menudo, al ejecutar un programa o un conjunto de programas juntos, la memoria disponible resulta insuficiente, y por ello los programas se conservan en un almacenamiento de apoyo de acceso rápido, como puede ser un disco rígido. Cuando se ejecutan los programas, el ordenador sólo carga las partes de un programa que se requieren para llevar a cabo la tarea de que se trate. Una vez acabada ésta, el programa se borra y se carga la siguiente parte del programa. Siempre y cuando el proceso sea suficientemente rápido, «parecerá» que todas las partes estén en la memoria y ejecutándose de forma simultánea. La aplicación práctica de la gestión de memoria virtual que se puede observar más fácilmente es en la multitarea, donde un ordenador ejecutará a la vez varios programas separados. Por ejemplo, podría ser necesario que un ordenador accediera a una carga de datos, los formateara y los imprimiera, manteniendo al mismo tiempo una red y facilidades para tratamiento de textos. Aquí, aunque parecerá que se están ejecutando los tres programas al mismo tiempo, habrá períodos en los que los programas estén «holgazaneando», a la espera de recibir, entrar o enviar mensajes desde un periférico. De este modo, los programas se pueden conservar en disco y cargar y ejecutar sólo cuando se los requiere. Mientras tanto, el ordenador puede utilizar la totalidad de su memoria para llevar a cabo otras tareas



Placa de 128 K de ampliación de memoria
El modelo estándar viene completo con esta placa de ampliación. No obstante, la memoria del PC/AT se puede ampliar a 3 Mbytes

Placas unidas de E/S en serie/en paralelo
Permiten la conexión de periféricos estándares en paralelo o en serie a través de puertas Centronics o RS232

Disco rígido de 20 Mbytes
Este componente del equipo ofrece almacenamiento masivo en línea

Placa adaptadora monocromática
Produce una visualización de gran calidad exclusivamente de textos

Placa para gráficos en color
Permite visualizar hasta 16 colores. También hay disponible una placa mejorada, que ofrece hasta 256 colores

Pila de litio
Esta pila de seguridad soporta un reloj de hora diaria y una pequeña zona de RAM estática que retiene, entre otras cosas, la configuración del sistema cuando se corta la alimentación de la red eléctrica

RAM en placa
El modelo básico incluye 256 K de memoria

Chips de ROM
Estos chips retienen el BIOS y las pruebas de diagnóstico tras la puesta en marcha para el sistema

Cerradura
Al girar esta llave el teclado queda cerrado, imposibilitando que se manipulen indebidamente las aplicaciones; impide, además, abrir la carcasa, evitando así el robo de las placas de ampliación

Altavoz
El PC/AT tiene incorporado un altavoz de 2 pulgadas

Procesador Intel 80286
Este procesador de 32/16 bits trabajando a 6 MHz le confiere al PC/AT su capacidad para «gestión de memoria virtual»

«Floppies» de 5¼ pulgadas
El PC/AT soporta discos de doble cara y gran densidad, que permiten el almacenamiento de hasta 1,2 Mbytes en un único disco

Chris Stevens

La suerte está echada

Con este capítulo finalizamos la construcción de nuestro tester digital y al mismo tiempo indicamos cómo comprobar las conexiones

Por las razones que explicamos ya en el capítulo anterior, se utilizan cinco LEDs idénticos. No es necesario utilizar zócalos para los LEDs, pero soldarlos directamente en las pequeñas patillas es bastante delicado. Si opta por emplear zócalos (que no están incluidos en la lista de componentes), recuerde que los zócalos DIL de 24, 28 y 40 patillas tienen la separación de patillas correcta (siete orificios en una placa matriz de 0,1 pulgadas).

Los LEDs se pueden montar en un trocito separado de placa matriz y conectarla a la placa principal mediante una corta manguera o un cable plano. Ello permitirá montar el visualizador en la ventana de una caja de instrumento pequeño. Las patillas 5 de los LED 3, 4 y 5 tienen conectados conductores que sirven para conmutar el punto decimal.

Se utiliza una fuente de alimentación simple, basada en dos reguladores de *tres terminales*. Esta

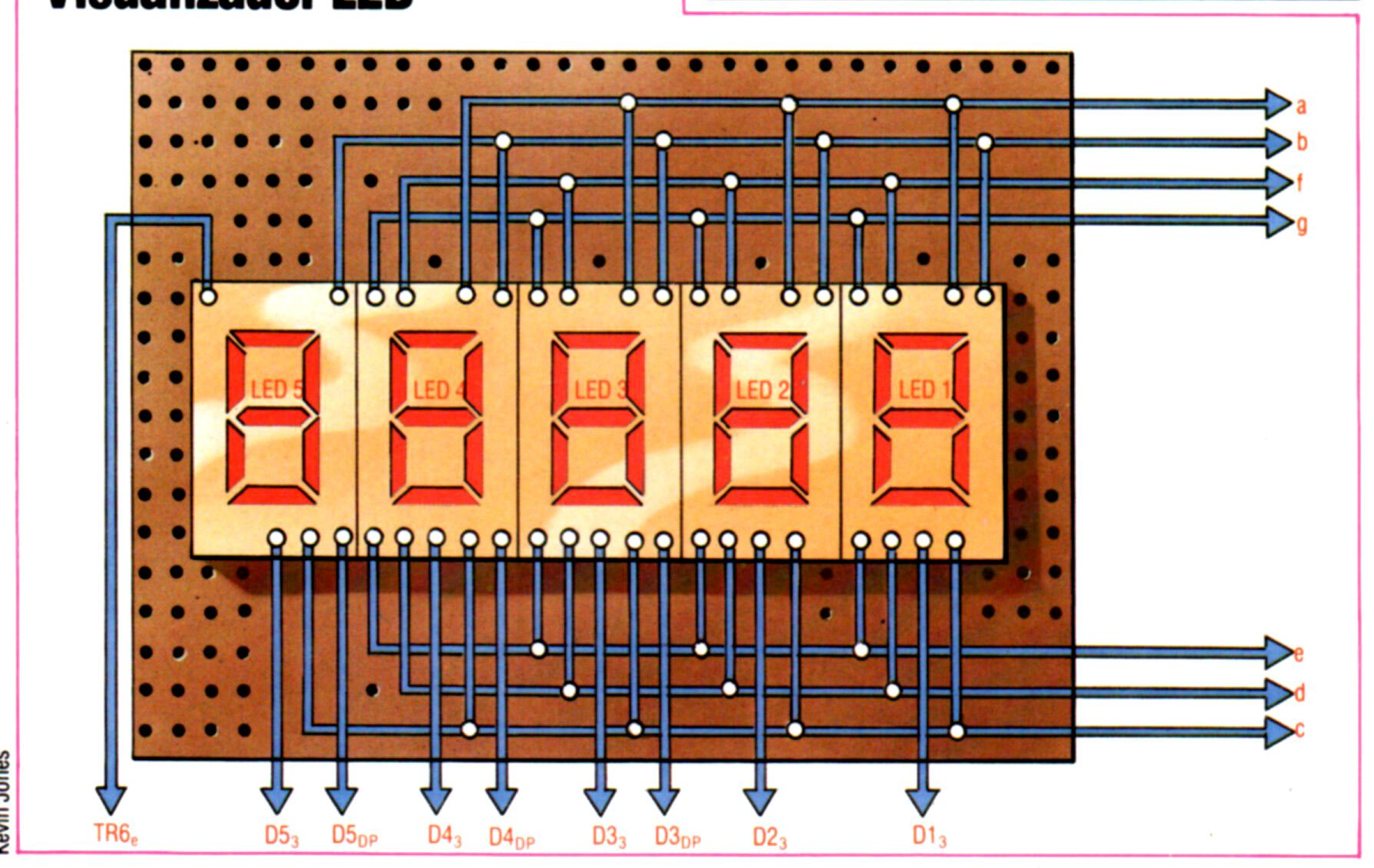
Lista de componentes

Cantidad	Ref.	Artículo
5	LED1-5	LED de siete segmentos de ánodo común
1	T1	transformador de 12 V
1	BR1	puente rectificador
1	C1	condensador electrolítico de 40 V 680 µF
1	C2	condensador de tantalio de 0,33 µF de perla
2	C3,C4	condensador de tantalio de 0,1 µF de perla
2	RG1,RG2	regulador de tres terminales de 5 V
1	—	disipador
2	D1,D2	diodo rectificador 1N4001
Varios		
1,5 m		hilo para arrollar
0,5 m		cable plano de 12 vías

El visualizador LED

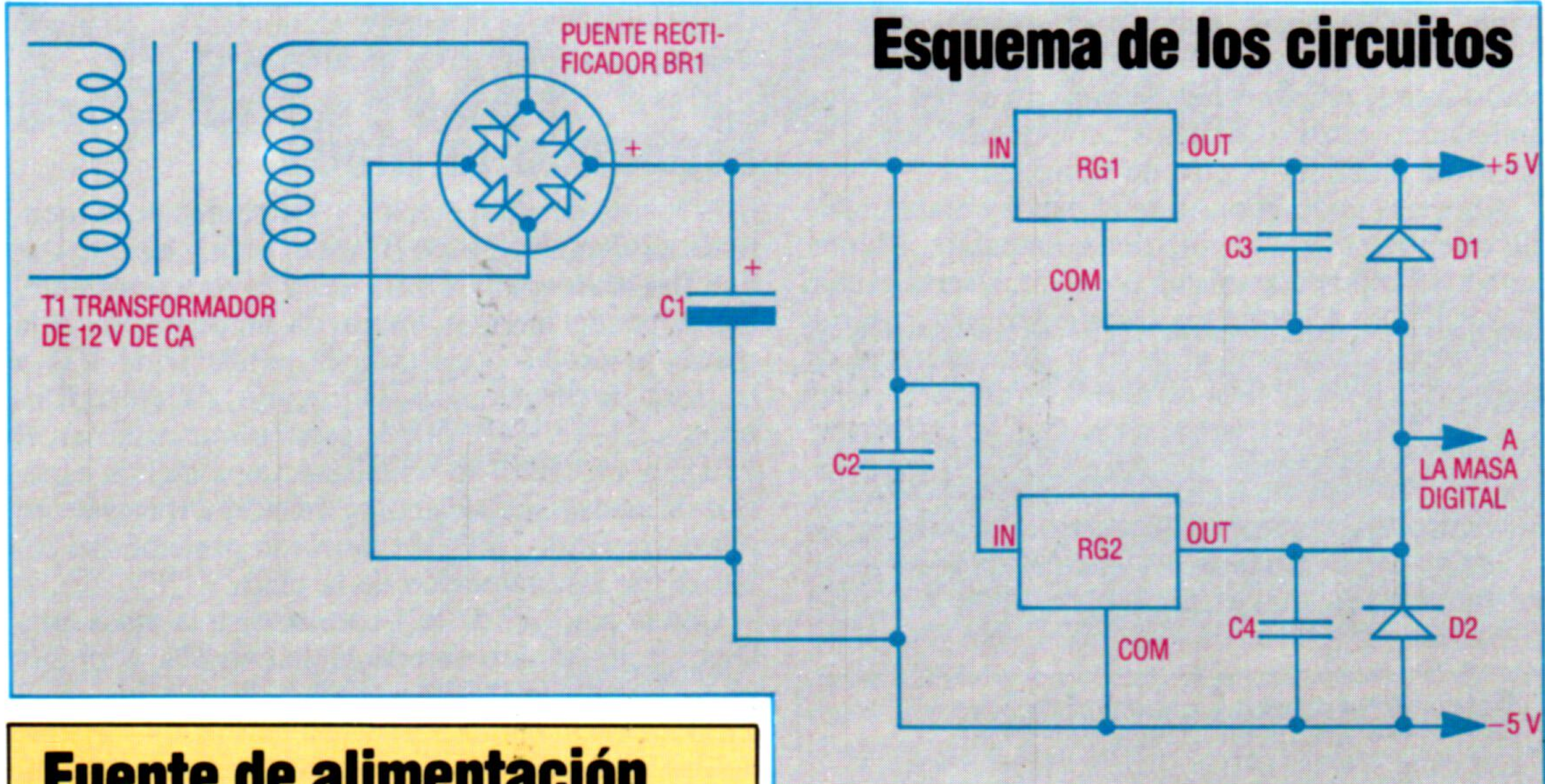
El de cinco dígitos para nuestro tester se monta en un trozo separado de placa matriz de 0,1 pulgadas (ver ilustración). Las siete señales que iluminan las barras LED (de a a f) son comunes para los cuatro visualizadores de dígitos situados más a la derecha, dado que estas señales se multiplexan entre los cuatro dígitos. Utilice el sistema de arrollamiento para realizar las conexiones de punto a punto necesarias. Luego se puede utilizar un trozo de cable plano para conectar estas siete líneas a la placa principal. Consultar la disposición de la placa principal para ver cómo se conectan estas líneas. Las señales de habilitación de dígitos desde la placa principal se conectan a la patilla 3 de cada visualizador LED. Nuevamente, remítase al diagrama de la placa principal. Leyendo este esquema desde la derecha, conecte $D1_3$ a D1, $D2_3$ a D2, $D3_3$ a D3, $D4_3$ a D4 y $D5_3$ a la conexión más sobre la derecha de los tres puntos marcados como D5. Vea que las conexiones del punto decimal, señaladas como Dn_{DP} y $TR6_e$ no se deben conectar en esta etapa, ya que serán el tema del último capítulo

Visualizador LED





Esquema de los circuitos



Fuente de alimentación

El circuito es muy directo, utilizando un puente rectificador y dos reguladores de tensión de tres terminales para convertir una corriente de entrada de 12 V de corriente alterna en alimentaciones de + 5 V y − 5 V para la placa principal. Los condensadores del circuito están para reducir «picos» (fluctuaciones súbitas) de la potencia dada por el transformador

configuración permite emplear una fuente única de 12 V de CA y un rectificador de un solo puente. No es necesario que los dos diodos sean del tipo especificado; servirá cualquier diodo rectificador pequeño. Éstos impiden una condición que se conoce como *enganche,* que podría producirse cuando se utilizan dos reguladores en una fuente de alimentación bipolar.

C1 puede ser un condensador electrolítico moderadamente grande, pero asegúrese de que posea una tensión de trabajo suficientemente alta (al menos 35 V). C2, C3 y C4 deben ser condensadores de tantalio sólido de gota, que toleran muy mal las tensiones inversas. Asegúrese de conectarlos con la orientación correcta: el «+» del cuerpo indica el lado positivo.

Los componentes de la fuente de alimentación (a excepción del transformador de red) se pueden montar convenientemente en una placa de tiras. RG2 sólo tiene que suministrar unos pocos miliamperios y no necesitará disipador, pero RG1 estará trabajando mucho más cerca de sus límites y, en consecuencia, habrá de atornillarse a un disipador pequeño. Tenga en cuenta que la cápsula de los reguladores de tres terminales está conectada al terminal central común. Si bien la mayoría de los disipadores son de aluminio anodizado y no son muy conductores (eléctricamente), es muy conveniente asegurarse de que el disipador no toque el metal del chasis ni ningún cable desnudo.

Una vez montada la fuente de alimentación, conecte el transformador de red a la red (tome las consabidas medidas de seguridad al hacerlo) y conecte una resistencia de 100 ohmios entre la salida de + 5 V y masa. Si tiene algún tipo de voltímetro (aunque sea uno barato de tipo analógico), mida la tensión en la resistencia, que deberá ser muy próxima a 5 V. Si dispone de un osciloscopio, ajuste la entrada de corriente continua a una sensibilidad de alrededor de 10 V. Aplique las puntas de prueba en los extremos de la carga ficticia de 100 ohmios y compruebe en la pantalla que no haya un rizado apreciable. Si lo hubiera, podría anular las prestaciones del DVM.

Los reguladores de tres terminales son dispositivos con los que es muy sencillo trabajar, pero pueden generar ruido de alta frecuencia. La disposición que vemos en la ilustración (al contrario de las conexiones indicadas en el esquema del circuito) ayudarán a minimizar este ruido.

Una vez probada la fuente de alimentación con resultados correctos, conéctela a la placa DVM principal, pero *no* inserte los IC en esta etapa. Utilice su tester para comprobar los + 5 V en la patilla 11 de IC1, en las patillas 4 y 8 de IC2 y en la 16 de IC3. Asimismo, compruebe los + 5 V en los colectores de TR1 a TR5, y los − 5 V en la patilla 1 de IC1. Ahora conmute a la escala de ohmios y conecte una punta de prueba del medidor al punto de masa analógica o digital. Ponga en contacto la otra punta de prueba con la patilla 3 de IC1, con la patilla 1 de IC2 y con la patilla 8 de IC3. Las resistencias que indique el medidor deberán ser muy bajas (de menos de 1 ohmio).

IC3 es un chip TTL corriente y se puede manipular sin ninguna precaución especial: tan sólo enchúfelo en el zócalo. Mientras tanto, asegúrese, sin embargo, de que la muesca del chip esté arriba, como se indicó en el último capítulo. IC2 e IC1 no se pueden tratar del mismo modo. IC2 es la versión CMOS del chip temporizador NE555, y se dice que está totalmente protegido contra descargas electrostáticas, pero es mejor no correr ningún riesgo. IC1 se destruirá sin duda alguna a causa de una manipulación incorrecta, y si con anterioridad usted no ha trabajado nunca con dispositivos sensibles electrostáticamente, a continuación le explicamos qué debe hacer.

En primer lugar, corte longitud adecuada de papel de aluminio de cocina y colóquelo sobre el banco. Luego desconecte la fuente de alimentación y coloque la placa del circuito en el centro de la hoja. Introduciendo la mano por debajo del papel, presiónelo firmemente contra el lado de las patillas del circuito. IC1 e IC2 se suministrarán con algún embalaje conductor. Éste podrá tener forma de es-

puma plástica negra, soporte de «goma» negra o soporte de plástico con tratamiento especial. Sea como fuera, independientemente de cómo vengan embalados estos dos chips, colóquelos sobre el papel de aluminio dentro de su embalaje.

Apoye su codo libre sobre el papel y manténgalo allí mientras libera al chip de su embalaje. Manteniendo el codo en la misma posición, inserte el chip en su zócalo. Cuando los chips estén bien insertados, ya puede retirar el codo y tirar el papel de aluminio. (Es muy fácil olvidarse de quitar el papel cuando conecte la fuente de alimentación, de modo que esté atento.)

¡Atención!

Todo proyecto que implique potencia eléctrica es peligroso. Desconecte la alimentación de la red antes de trabajar con la placa

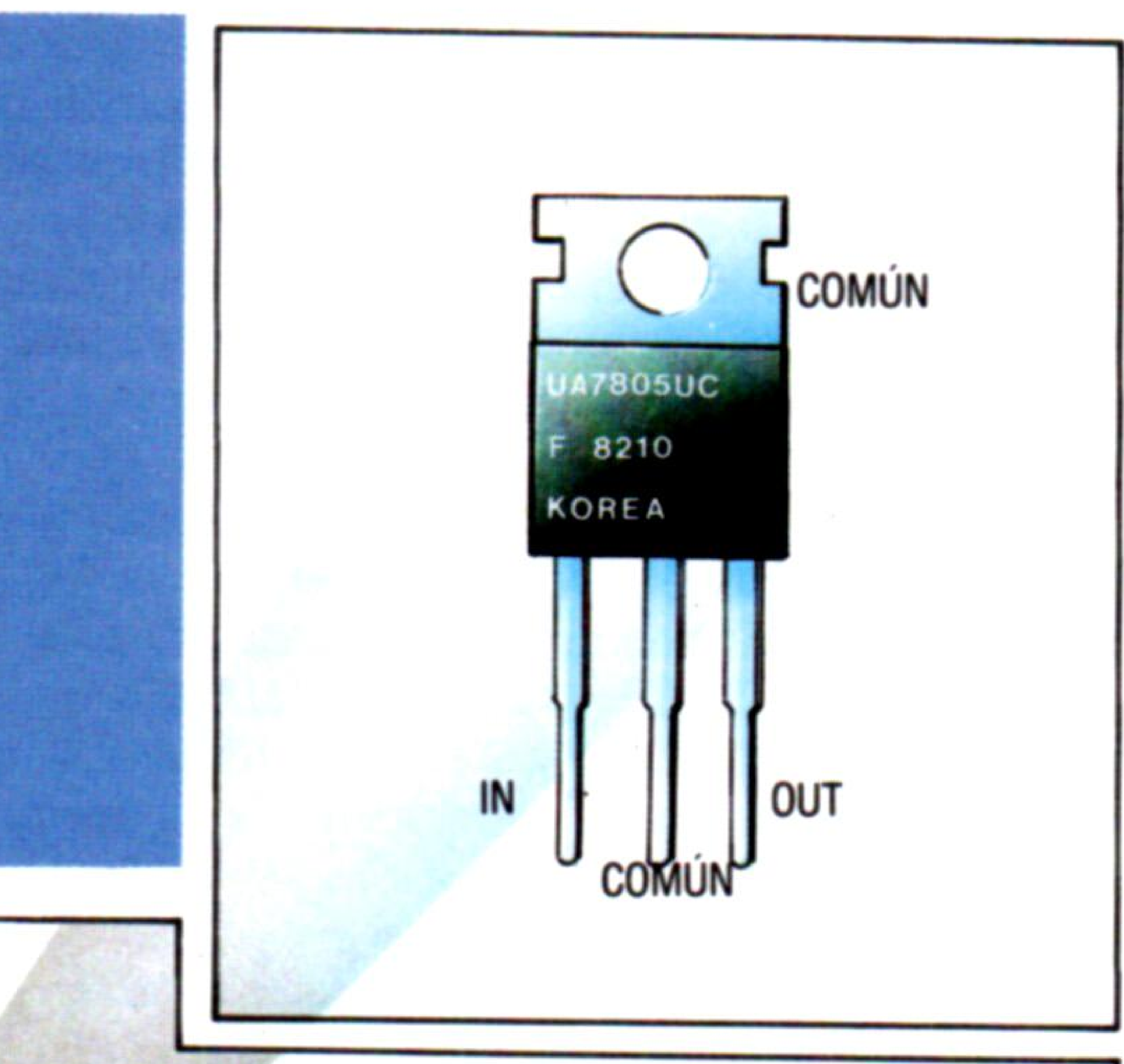
Montaje del regulador

Los componentes para el circuito de rectificación y regulación de la fuente de alimentación se deben montar sobre un trozo de placa de 36 tiras. Este tipo de placa es ideal para circuitos de esta clase. Suelde los terminales de los diversos componentes a las tiras, como se indica, y emplee cables aislados para unir las tiras entre sí. La entrada de corriente alterna del transformador se debe conectar a la placa principal del tester. La alimentación de −5 V se conecta a un punto próximo a la parte inferior de esta placa, y la línea de masa se conecta al punto principal de la masa analógica de la placa. Dado que la alimentación de +5 V ha de alimentar varios puntos de la placa principal, le aconsejamos que monte este cable alrededor del borde de la placa principal y lo derive donde resulte conveniente. Es importante observar la orientación de los condensadores y diodos, el puente rectificador y los reguladores. En especial, se deben insertar correctamente los reguladores de tensión de tres terminales. Para determinar la orientación, ponga el regulador sobre su cara posterior, de modo que los caracteres impresos del cuerpo queden arriba. De izq. a der., las patas son IN (entrada), COMÚN y OUT (salida)

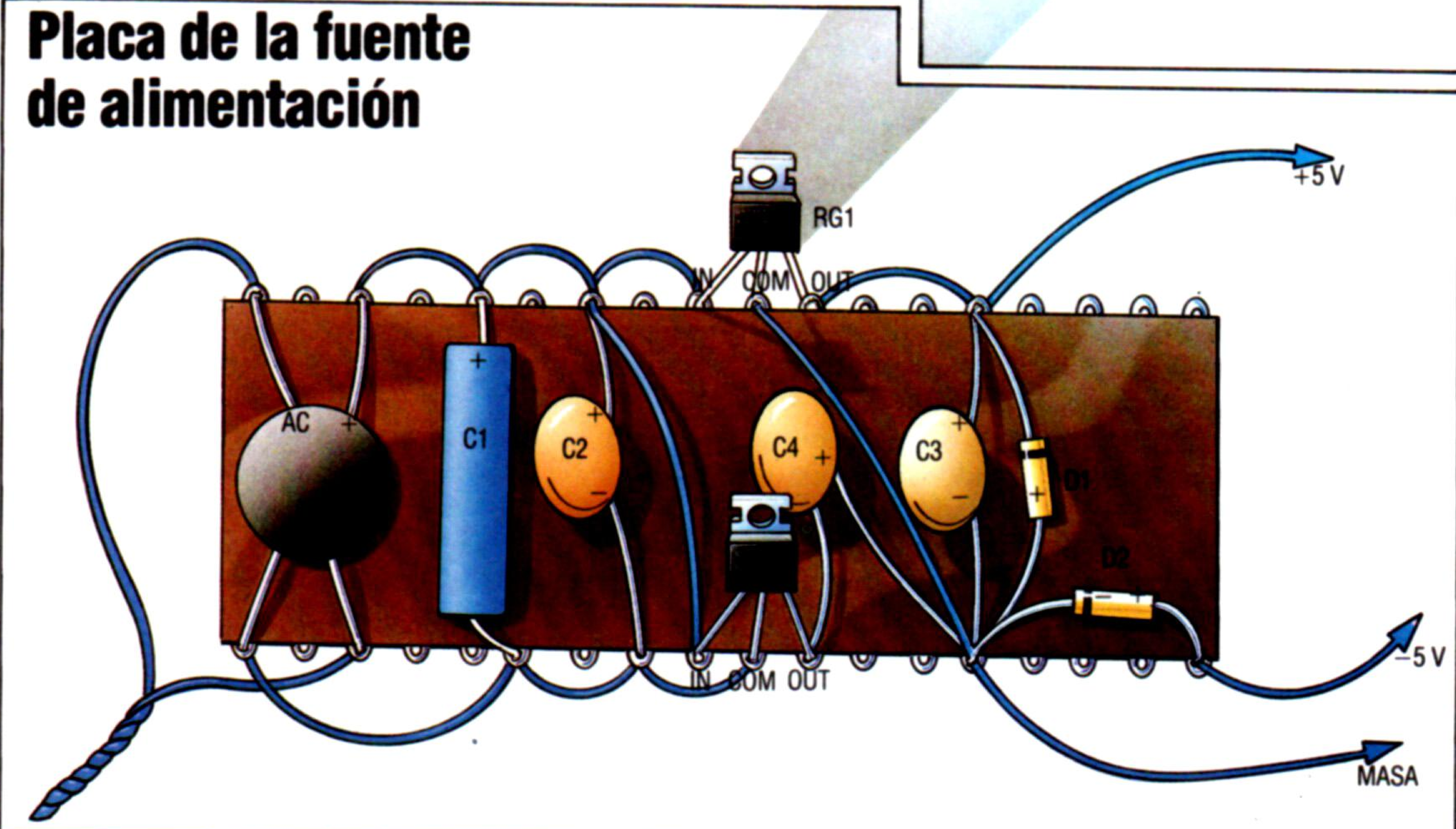
Probando la placa

Si la fuente de alimentación y los puntos de la fuente de alimentación a los IC están bien y los chips se han insertado de la forma descrita anteriormente, ahora puede intentar aplicar la alimentación a la placa. Si no ha conectado la patilla 9 de IC1 a la masa analógica, con un circuito abierto en los puntos IN− e IN+, puede esperar ver cómo en el LED1 y el LED2 se visualizan unos dígitos espurios. Si estos dígitos efectúan un ciclo a través de un patrón repetido, trate de mover la posición del cableado de interconexión de la placa.

Con la patilla 9 de IC1 conectada a la masa analógica y un circuito abierto en la entrada, verá una visualización de 00000 o un 1 o 2 espurios en el LED1. Con la patilla 9 de IC1 a masa e IN− e IN+ unidas, obtendrá una visualización estable (sin intermitencias) de 00000 o 00001. Si usted ha tenido que transigir con D1 (el diodo de referencia de tensión) o C1 (el condensador integrador) quizá no obtenga un rendimiento inicial tan satisfactorio. En cualquier caso, una visualización que efectúe ciclos indica que la disposición del cableado es deficiente, de modo que deberá modificar la disposición de los conductores hasta que la visualización sea estable.



Placa de la fuente de alimentación

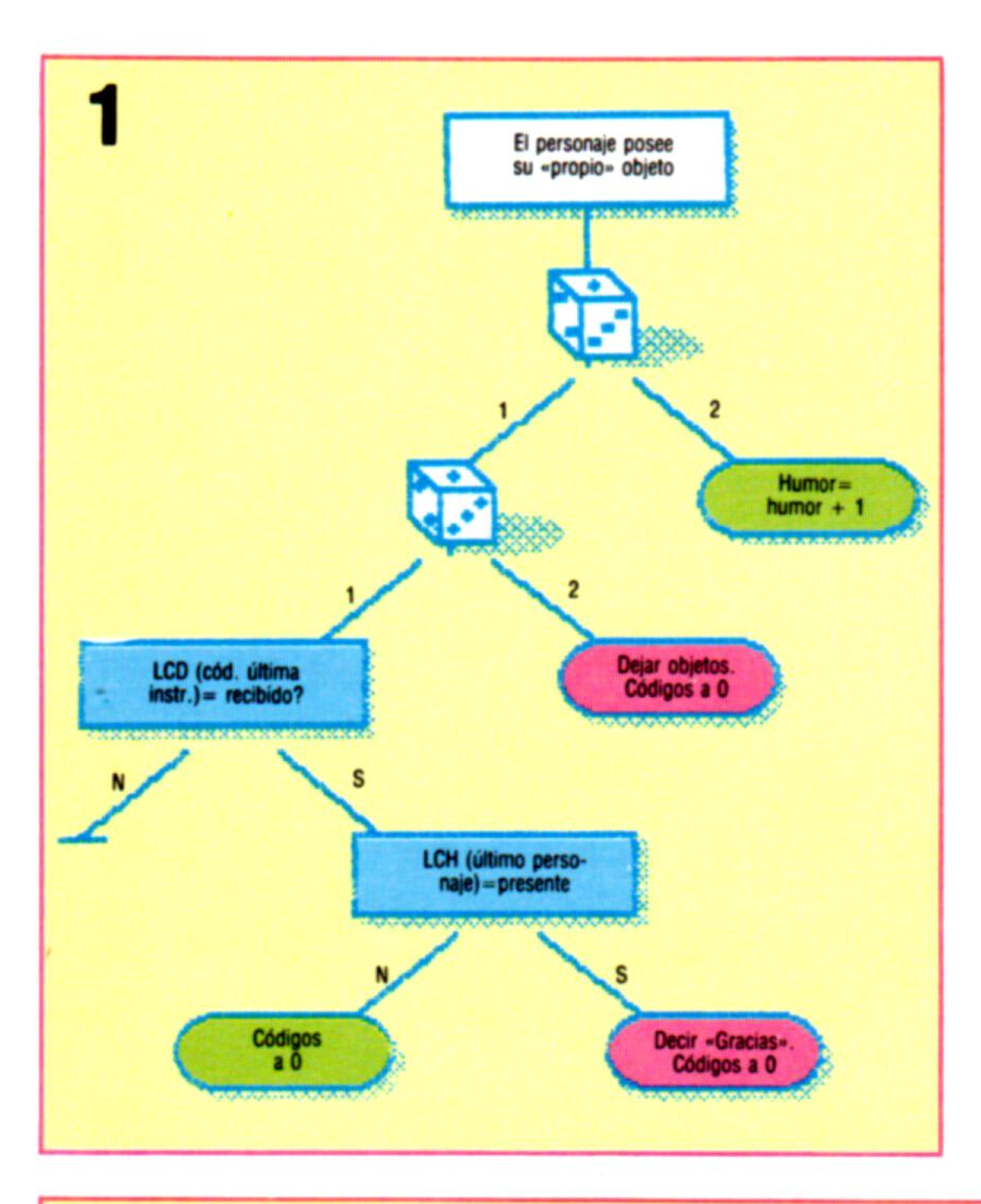


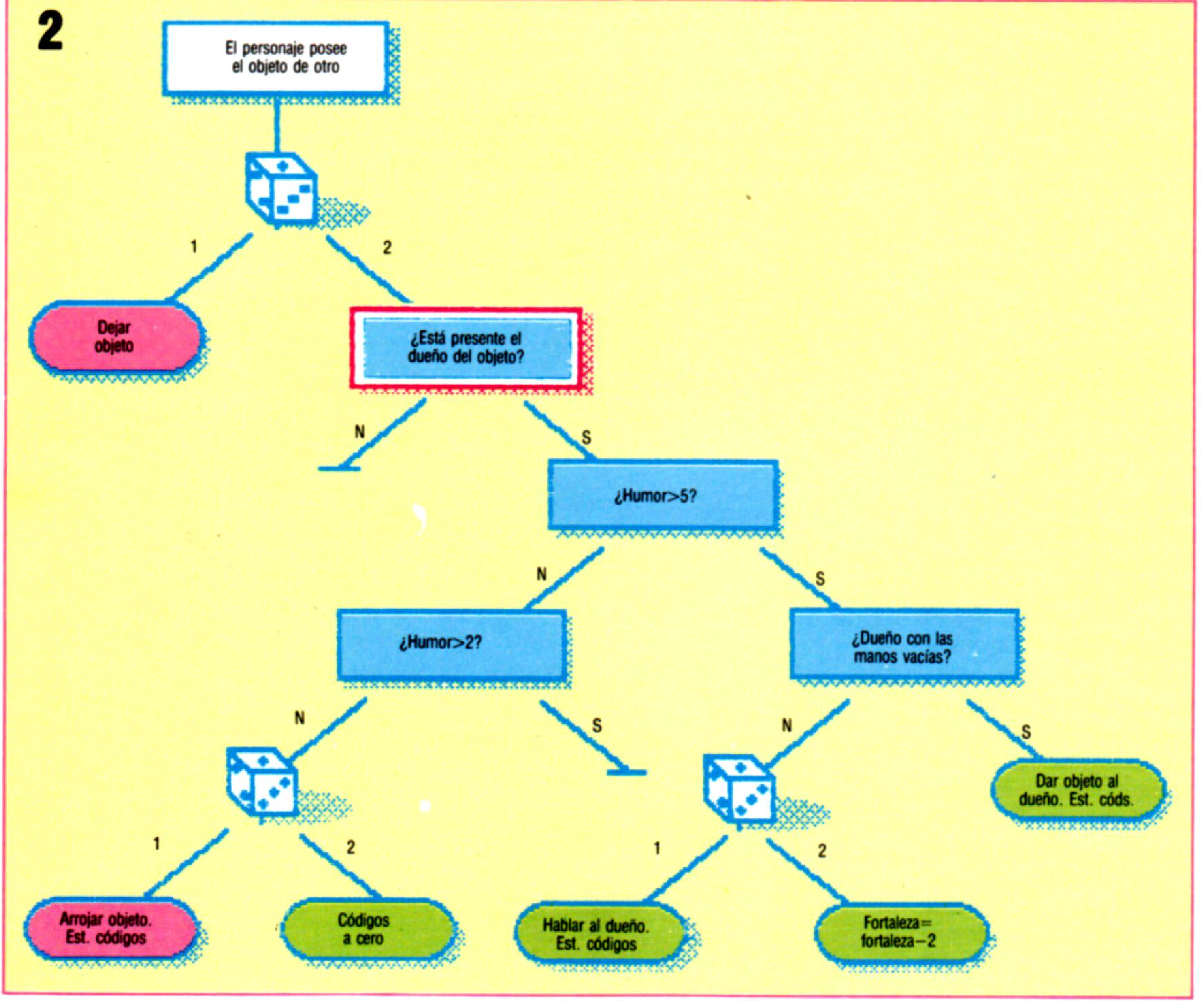
Lección «objetiva»

La manipulación de objetos es esencial en los juegos con personajes interactivos

La forma en que nuestros personajes manipulan los objetos es muy importante en el Dog and Bucket, y se deben tener en cuenta una gran cantidad de factores. En primer lugar, los objetos se integran en dos categorías principales: comestibles (el bocadillo y la empanada) y bebestibles, pero también se incluyen tres objetos (el cenicero, la banqueta y la lata) cuya finalidad es muy limitada, aparte de contribuir a «crear atmósfera». Sin embargo, como veremos, tiene implicaciones en la trama.

El otro atributo fundamental de los objetos es su pertenencia. Ciertas bebidas pertenecen, por ejemplo, a ciertos personajes, y es obvio que necesitaremos llevar el registro de las posesiones de cada uno. Igualmente importante es el hecho de que tendremos que tener en cuenta lo que desea cada personaje. Todos estos factores se pueden comprobar mediante estructuras arborescentes similares a al-





gunas que ya hemos visto con anterioridad, aunque más complejas. Para simplificar las cosas, abordaremos la cuestión de los objetos en cuatro etapas.

En primer lugar, consideremos los posibles cursos de acción que puede seguir un personaje que se halla en posesión de un objeto.

El primer diagrama muestra una estructura arborescente simple para tratar esta eventualidad. Entre los puntos a destacar se incluyen los símbolos de los dados, que representan *nudos aleatorios* cuyo valor (que determinará el nudo hacia el cual se dirigirá la siguiente bifurcación) se fija al azar cuando se encuentren en el recorrido del árbol.

Además de nudos aleatorios, en el diagrama hay otros tres tipos de nudos. Los rojos son nudos terminales, en donde se imprimirá un mensaje y se actualizarán las variables. Los nudos verdes también son nudos terminales, pero éstos sólo actualizarán variables sin alterar la visualización, de modo que cuando entremos el árbol en nuestro programa podremos ocuparnos de ellos por separado. Por último, los nudos azules son simples nudos de elección, cada uno de los cuales retendrá un valor condicionado y se bifurcarán en consecuencia.

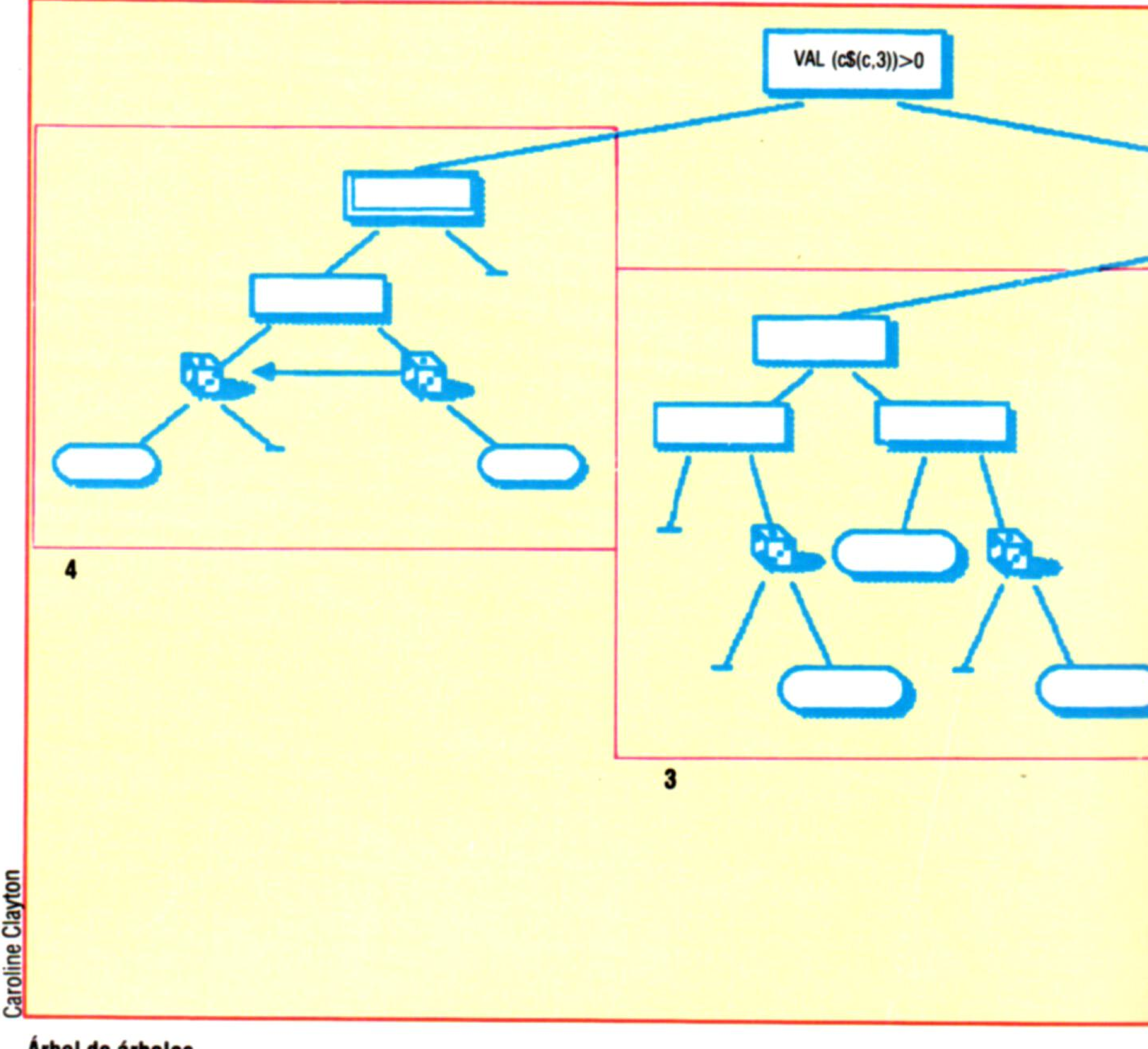
A partir de este diagrama usted puede ver que cuando un personaje posee un objeto «propio», es posible uno de cuatro resultados, representados por los cuatro nudos terminales. Existe la posibilidad aleatoria de que mejore el humor del personaje, en virtud de poder abandonarse a su bebida favorita. Por el contrario, aunque las probabilidades son menores, el personaje podría dejar el objeto o bien, si acaba de recibirlo y quien se lo entrega está todavía presente, decir Gracias. Observe que tres de los nudos terminales resultan en el establecimiento en cero de los códigos LCH y LCD, con lo cual se da por terminada cualquier interacción previa con otros personajes.

Las estructuras como éstas son puramente arbitrarias, lo que confiere la libertad de diseñar árboles bastante diferentes para los mismos fines. Teniendo esto presente, pasemos al segundo diagrama, que muestra qué podría suceder cuando el personaje en cuestión ha entrado en posesión del objeto de algún otro personaje.

Nudos del segundo árbol

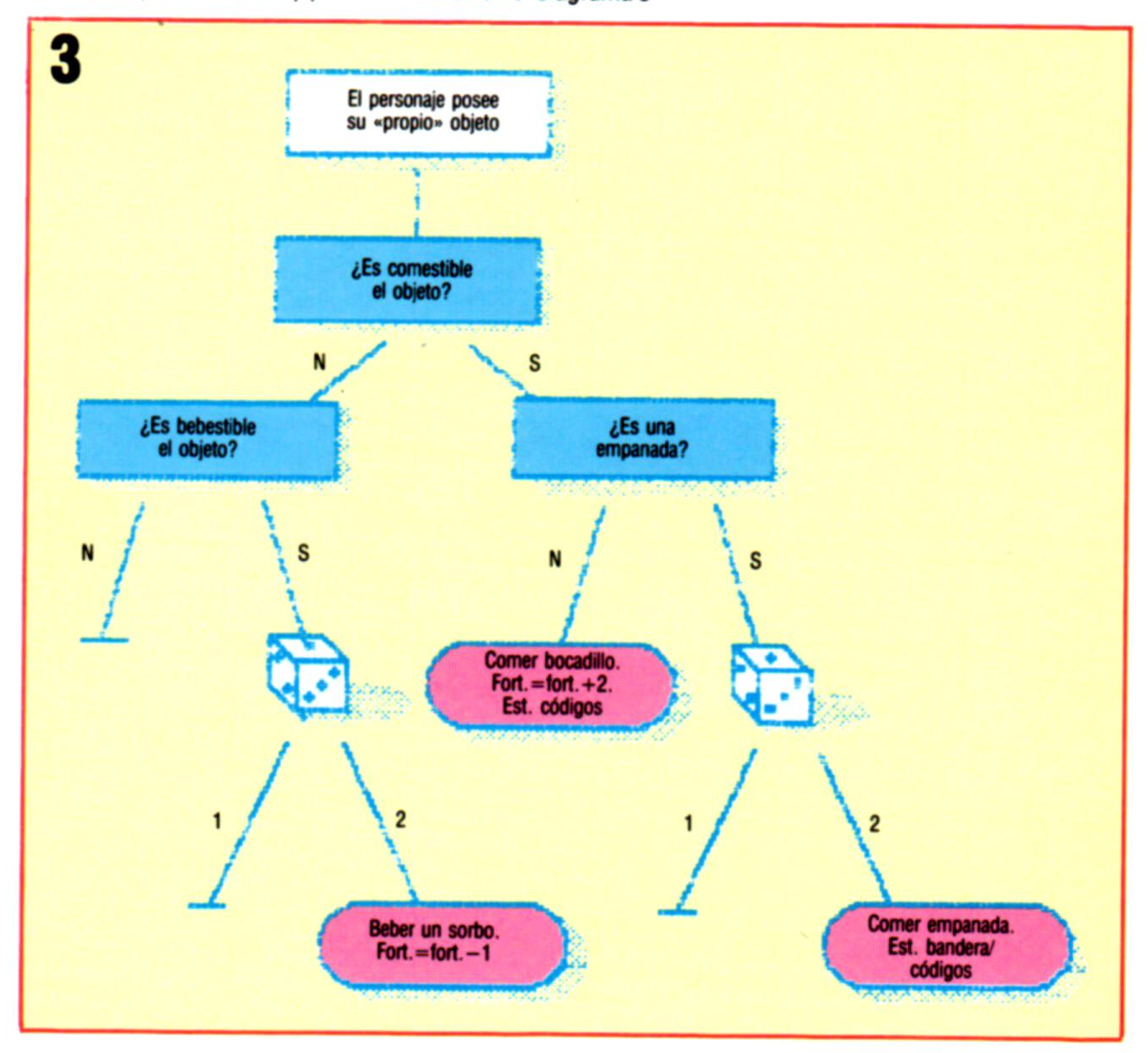
Los nudos de este árbol poseen un código de color de estilo similar a los del árbol anterior, con la excepción de un nudo enmarcado en rojo. Éste actúa de modo similar a un nudo terminal, de modo que, al llegar al mismo, el programa saltará fuera del árbol hasta una rutina situada en algún otro lugar del programa. No obstante, una vez ejecutada dicha rutina, y en función del resultado obtenido, el programa podría volver a saltar al árbol y continuar su descenso a través del mismo.

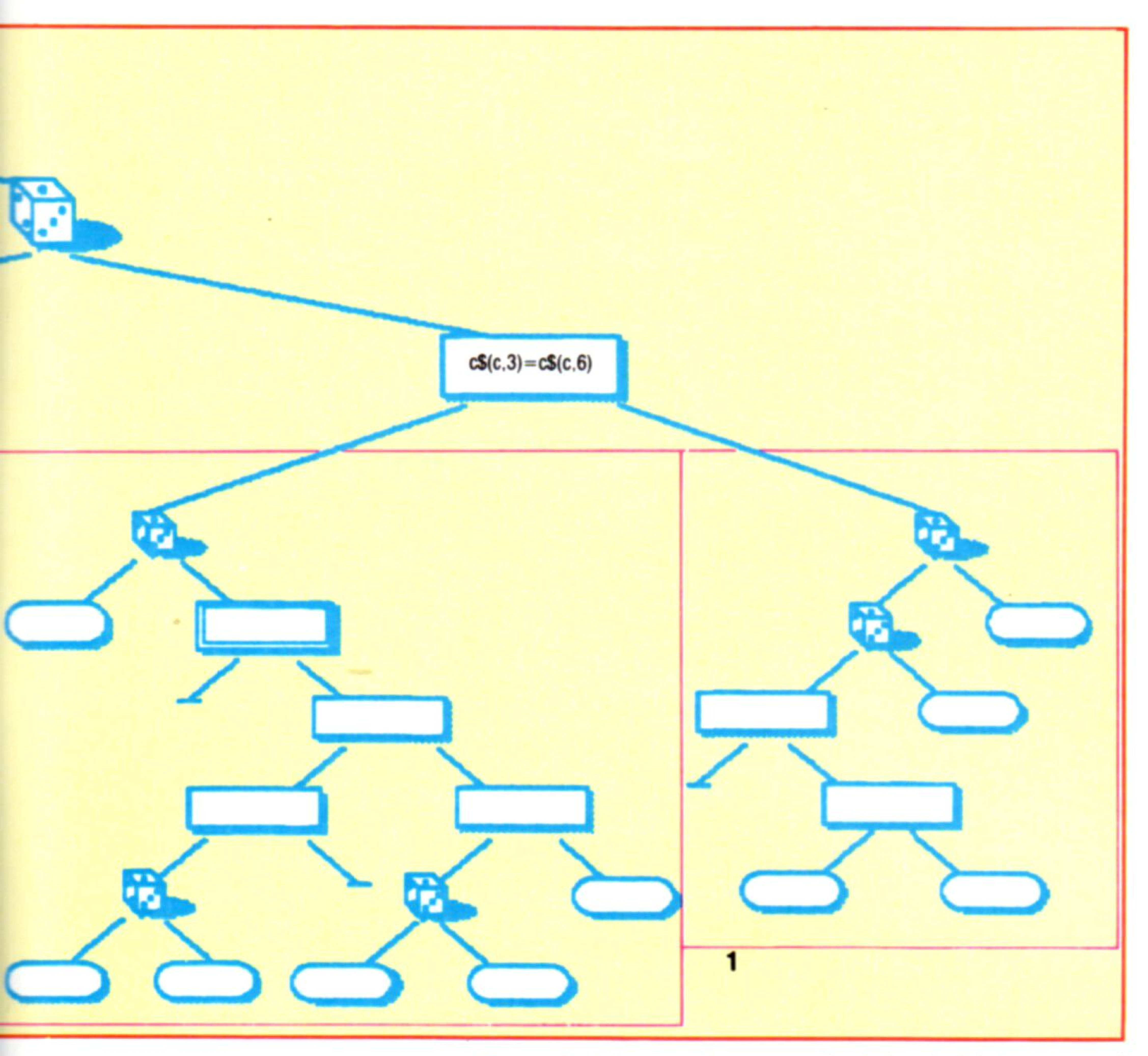
Para averiguar si el propietario del objeto retenido por nuestro personaje se halla en la misma habitación, necesitamos pasar por los atributos de escenario (c$(personaje,2)) de cada persona. En este punto, el nudo en cuestión nos empuja fuera del árbol, a una rutina sencilla que hace esto por nosotros. Si el personaje en cuestión está presente, dará un salto hacia atrás en el árbol comenzando a partir del nudo siguiente (probando humor>5); pero si está ausente, se abandona el descenso y volvemos hasta donde habíamos comenzado.



Caroline Clayton

Árbol de árboles
Las cuatro estructuras arborescentes que hemos explicado en este capítulo se pueden combinar entre sí para formar un árbol grande, como podemos ver arriba. El programa comprueba primero el valor de la expresión (VAL(c$(c,3))>0), cuya evaluación dará VERDADERO si el personaje en cuestión (c) tiene consigo un objeto, y FALSO en caso contrario. De dar FALSO, el flujo del control cae hacia abajo del árbol principal hasta la sección que se puede apreciar (con mayor detalle) en el diagrama **4.** De dar VERDADERO, el programa toma una decisión aleatoria respecto a si recorrer la estructura que se indica en el diagrama **3** (comer/beber objetos) o bien seguir adelante y comprobar los objetos «propios» mediante la evaluación de la expresión (c$(c,3)=c$(c,6)). El control pasa luego a una de las dos estructuras que se reflejan en los diagramas **1** y **2**





Es probable que el personaje deje el objeto; pero si no sucede esto, el programa comprueba la presencia del dueño. En este punto, si éste está cerca, el programa decide que el objeto se le debe entregar (siempre que el receptor esté con las manos vacías) o bien arrojar al propietario, si el personaje se halla de un humor particularmente malo. Si el dueño tiene las manos ocupadas y, en consecuencia, es incapaz de recibir el objeto, existe la posibilidad aleatoria de que el personaje diga: Lo siento, creo que he tomado tu bebida..., o algo por el estilo.

La única otra cosa que debemos observar es que uno de los nudos terminales implica la reducción de la fortaleza del personaje en dos puntos, dado que existen todas las posibilidades de que el personaje beba unos sorbos del vaso si es incapaz de devolvérselo a su propietario original; y, como todos sabemos, ¡cuando se mezclan bebidas los efectos son fatales! Sin embargo, aparte de esto, ninguno de los árboles que acabamos de describir hace ninguna provisión para comer o beber. En consecuencia, necesitamos una estructura que se ocupe de esto, que es la que se muestra en el tercer diagrama.

Este árbol tiene un código de color similar a los dos anteriores y, nuevamente, la fortaleza de un personaje se puede reducir en función de la bebida, si bien no en tanta cantidad, dado que en este punto no hay ninguna prueba para ver si se está muestreando la bebida correcta. Comer el bocadillo aumenta en dos la fortaleza del personaje, y establece el LCD de la forma correspondiente. Sin embargo, comer la empanada establece una bandera, reservada a los fines del argumento.

Y, ya que ha surgido el tema, bien podríamos hacerle conocer algunos secretos y revelarle la trama del argumento de Dog and Bucket. Sin lugar a dudas, se habrá preguntado el significado de las latas de alimento para gatos de la cocina, y habrá adivinado que las famosas empanadas de carne del Dog and Bucket en realidad están elaboradas con un alimento para animales de conocida (pero barata) marca.

4

El personaje no posee objetos

¿Hay algún objeto presente?

S N

¿Están presentes algunos objetos?

N S

1

1 2 2

Selec. y coger objeto. Est. códs.

Coger objetos. Est. códigos

Veneno en la comida

Comer una empanada tendrá como resultado el fallecimiento intempestivo de uno o más de nuestros personajes, cuya identidad exacta diferirá cada vez que se ejecute el juego. En consecuencia, necesitamos establecer una bandera que indique quién, si hay alguno, sufrirá probablemente un envenenamiento alimentario. El hecho de que la muerte de un personaje sea considerada por los otros clientes del *pub* como un misterio o un asesinato, dependerá de otros factores, de los que nos ocuparemos en el próximo capítulo. Mientras tanto, sigamos examinando las estructuras de control de los objetos.

En caso de que un personaje tenga las manos vacías, utilizaremos el árbol esbozado en el cuarto diagrama, en el cual hay otro nudo que nos saca del árbol y nos lleva a una subrutina. Esta rutina comprueba si en la habitación hay otros objetos presentes, retornando si los hubiera y comprobando luego si el «propio» objeto del personaje está o no presente. Si lo está, hay una opción aleatoria de recogerlo; pero si no está, existe la posibilidad de recoger otro objeto. Tenga en cuenta que si los personajes no recogen sus «propios» objetos, existe otra posibilidad de que recojan otros, porque el nudo en cuestión salta a través del árbol, dando lugar, por tanto, a esta posibilidad. Esto significa que no es probable que los objetos se dejen en un sitio durante mucho tiempo, pero que los personajes *tenderán* a recoger sus propias bebidas.

Por último, se pueden unir estos cuatro árboles para conformar una estructura grande, y cada una de las condiciones que es necesario comprobar se puede expresar como una única expresión lógica. Puesto que algunos nudos querrán probar las mismas condiciones en distintos momentos, les asignaremos los valores de las sentencias condicionadas a una nueva matriz, utilizando elementos de ella (cuando fuera apropiado) para construir estructuras arborescentes.

Contra reloj

La temporización es una de las características más importantes de un OS. Veamos cómo se pueden tratar las interrupciones y los eventos, los dos métodos de temporización del Amstrad

La temporización del sistema gobierna cualquier operación que haya de efectuarse en un instante dado: por ejemplo, los colores parpadeantes de las letras en la pantalla deben ejecutarse durante un período de *retorno de cuadro* (el tiempo necesario para actualizar la visualización en video). Hay dos métodos distintos para implementar esta temporización en los ordenadores Amstrad; el primer método es controlado por el hardware y se denomina *interrupción*, y el segundo es controlado por el software y se denomina *evento*. Trataremos el empleo que hace el sistema de las interrupciones, como inicio de nuestro análisis.

Una interrupción es una señal generada fuera del microprocesador Z80, el cual suspende la ejecución del programa actual durante el tiempo en que se trata la señal.

Los ordenadores Amstrad operan con el Z80 programado para interrupciones del modo 1, es decir, se pasa el control a la rutina encargada de tratar las interrupciones, residente en la memoria en $0038. En teoría, una interrupción puede ser generada en cualquier momento y por cualquier dispositivo que reclame la atención del Z80, aunque en un Amstrad no ampliado sólo puede producirse un tipo de interrupción, conocido como *interrupción de temporizador* (*timer interrupt*).

Interrupciones del sistema
El procesador del Amstrad puede ser interrumpido por el sistema operativo de cuatro maneras. Estas interrupciones de temporizador se producen a intervalos regulares, para sincronizar las funciones regulares del sistema operativo, tales como el rastreo del teclado, la actualización del visualizador del video y el control de la transferencia de datos al chip de sonidos del Amstrad

Interrupción rápida 1/300AVO SEG.
Int. para generación sonido 1/100AVO SEG.
Interrupción de reloj 1/50AVO SEGUNDO
Int. de retorno cuadro 1/50AVO o 1/60AVO SEG.

La interrupción de temporizador se genera directamente desde el reloj del sistema (que funciona a 4 MHz) por la ULA, y sucede cada 300-ésima de segundo.

Toda temporización interna (tal como la que necesita el BASIC para sus instrucciones AFTER y EVERY) depende de esta interrupción.

La interrupción de temporizador se trata directamente por una parte del firmware denominado *kernel* (núcleo) que dialoga con el resto del firmware señalando diferentes niveles de interrupción, conforme se muestra en el esquema «Interrupciones del sistema». Veamos cada una de estas cuatro señales por separado:

- La *interrupción rápida* se produce cuando el núcleo recibe una interrupción de temporizador.
- La *interrupción para generación de sonido* se produce cada tres interrupciones de temporizador, y es empleada por el firmware para sincronizar la transferencia de los datos al chip de sonido. El usuario no puede detectar cuándo ocurre esta interrupción, salvo si se mantiene un contador de las interrupciones rápidas.
- La *interrupción de reloj* se produce después de seis interrupciones de temporizador y es empleada por el firmware para iniciar un rastreo de pantalla.
- La *interrupción de retorno de cuadro* se produce después de cada cinco o seis interrupciones de temporizador (según el tipo de visualización video) y se emplea para toda temporización dependiente de la pantalla.

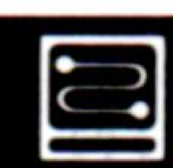
El firmware mantiene un contador de interrupciones rápidas que el usuario puede obtener, y permite el establecimiento de bucles de temporización sin tener que suspender las actividades de un programa. El contador puede establecerse llamando KL_TIME_SET, que necesita que el valor para el temporizador sea pasado al par de registros HL. El valor del contador actual puede ser determinado en cualquier momento llamando KL_TIME_PLEASE, la cual da el valor contenido en HL. Un programa que sirva de ejemplo sobre el uso de estas dos rutinas es el que ofrecemos en estas páginas; el programa sondea el teclado durante un intervalo de tiempo especificado antes del retorno. Puede incorporarse a un programa que permita al usuario un determinado tiempo de respuesta antes de ejecutar una operación (como visualizar un mensaje de ayuda).

Se pueden emplear directamente las interrupciones de temporizador parcheando la entrada RST 7 en $0038. El código parcheado primero debe desalojar y copiar en otro sitio la entrada existente de tres bytes, y después cargar allí la nueva entrada (que normalmente será un salto a una dirección en la memoria). Esta nueva entrada deberá ser reubicable, de modo que si es desalojada a su vez por una segunda rutina, pueda funcionar correctamente aun en este caso.

La nueva rutina puede realizar cualquier tarea que se le pida (p. ej., actualizar el reloj) y deberá ejecutar entonces la entrada original saltando a la posición en que fue guardada. Este procedimiento asegura que todavía sean ejecutadas cualesquiera funciones de interrupción de temporizador, tales como sondear el teclado. La idea de insertar una *cuña* de código en una rutina de interrupción del sistema operativo ha sido ya analizada en capítulos anteriores de este curso de lenguaje máquina.

Interrupciones externas

Hasta ahora nuestro estudio ha supuesto que toda interrupción recibida por el núcleo era generada por la ULA, pero es posible que un periférico externo genere una interrupción. Por ejemplo, una interface serial puede generar una interrupción siempre que se haga obtenible un carácter; en este caso una rutina habrá de ser proporcionada para que lea el carácter desde la interface.



Para que esto sea posible, el núcleo debe ser capaz de distinguir entre una interrupción de temporizador y una interrupción externa. Normalmente, cuando se detecta una interrupción, el Z80 se deshabilita para recibir cualquier otra interrupción (y esto es para hacer que no sean violadas otras interrupciones de temporizador). El núcleo activa interrupciones desde dentro de la rutina que trata las interrupciones, y si la señal de interrupción al Z80 es todavía baja (indicando una petición de interrupción), entonces se genera una segunda interrupción. Dado que no se duplica una interrupción de temporizador a este punto, esto puede servir para detectar una interrupción externa siempre y cuando la señal de la interrupción permanezca baja hasta ser atendida. Cuando se detecta una interrupción externa, el control pasa a $003B, por lo que esta posición debe ser acoplada con un salto a la rutina de servicio de interrupciones, la cual atiende al dispositivo que la solicita.

El hardware de sistema acepta tanto *interrupciones no enmascarables* (NMI: *non-maskable interrupts*) como interrupciones estándar. Las NMI sólo difieren en que no pueden ser deshabilitadas en cualquier momento, y su empleo no es recomendable porque puede afectar a actividades de sistema con peticiones limitadoras de temporización. En particular, las secciones firmware de tratamiento de disco y cassette (y partes de las secciones de tratamiento de sonido) continúan operando con interrupciones deshabilitadas, pero quedarían seriamente afectadas si se usaran NMI.

Eventos software

Al nivel más sencillo, los *eventos* pueden ser descritos como el equivalente en software de las interrupciones: las tareas particulares son realizadas por separado respecto de cualquier programa principal por medio de un contador de eventos a ejecutar (cuáles y cuándo) que guarda el firmware.

Aunque los eventos pueden ser empleados para tratar las interrupciones externas, son de gran utilidad para programas complicados que incluyan una simulación, o que requieren una operación independiente del programa principal que aparece en primer plano. Por ejemplo, el sintetizador de voz SSA-1 de Amstrad utiliza un evento para permitir que se genere la voz de manera transparente desde dentro de un programa en BASIC.

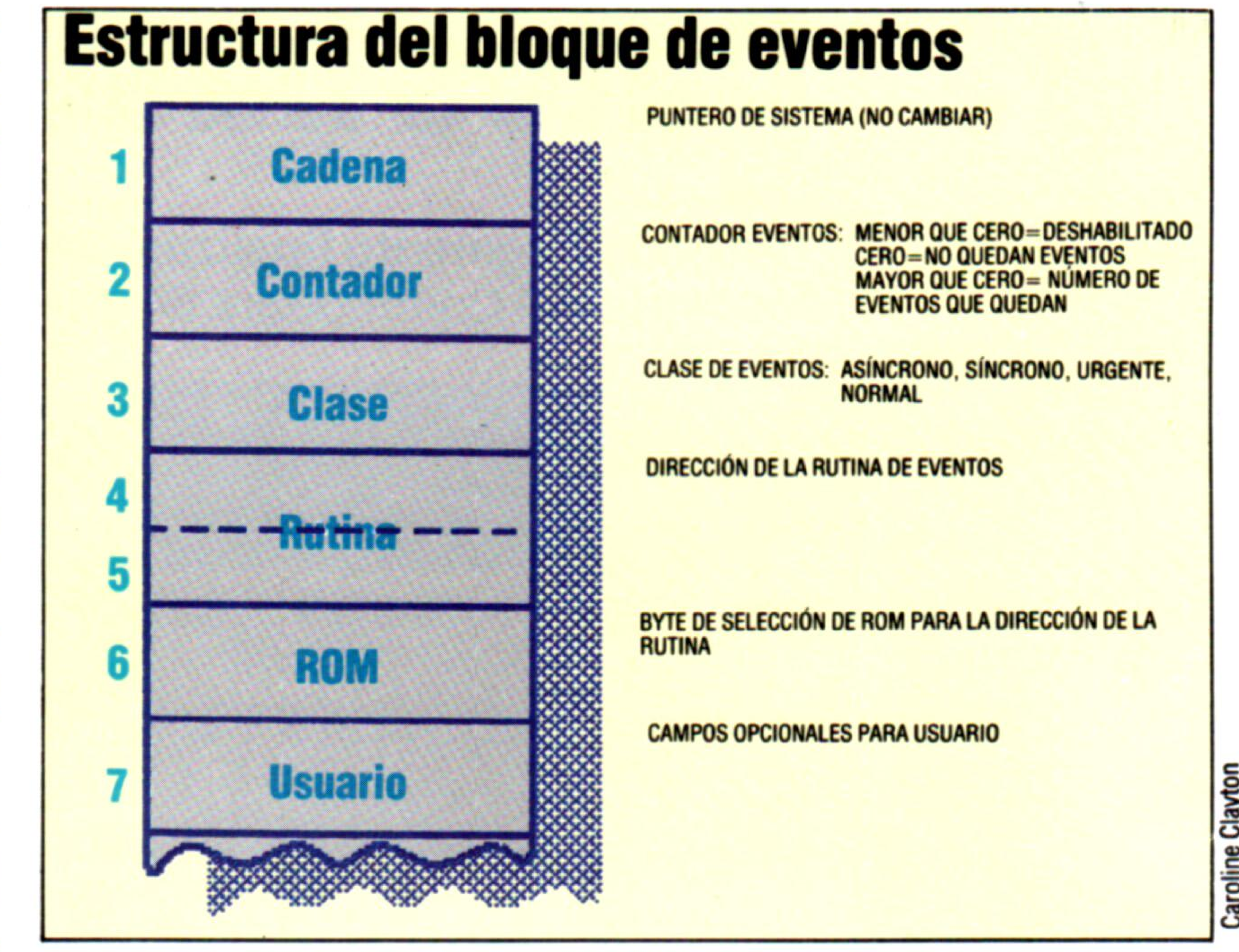
Los eventos se presentan al sistema operativo por medio de un *bloque de eventos*. Éste consiste simplemente en un bloque de siete bytes contiguos situados en algún lugar dentro de los 32 K centrales de la memoria. Un bloque de eventos tiene la estructuración que muestra nuestro diagrama; dentro del bloque existen tres campos, esenciales a todo evento, que caracterizan el tipo de evento y dónde ha de ser procesado.

Dentro del bloque de eventos está el *contador de eventos*, que cuenta las veces que ha ocurrido un evento. El proceso de incrementación de este contador es conocido como *puntapié* (*kicking*). Cada vez que se lanza un evento el contador es incrementado en una unidad; una vez procesado el evento con la llamada a la rutina de eventos, el contador es disminuido en una unidad. Un modo de neutralizar un evento es darle al contador un valor negativo.

La *clase de evento* indica si éste es *síncrono* o *asíncrono* con relación al programa principal. Los eventos asíncronos son procesados independientemente del programa principal y se destinan a aplicaciones que requieren casi una acción inmediata.

Los eventos síncronos están situados en una cola cuando se producen, de tal modo que puedan ser procesados por el programa principal cuando sea necesario. Los eventos síncronos son procesados conforme al orden en que ocurren. Además, el evento puede ser también *urgente* o *normal*.

En el capítulo próximo consideraremos las diferencias entre las varias clases de eventos y el modo que tiene el sistema operativo de tratarlos. Puede que a usted le parezca que el empleo de eventos presenta un buen número de complicaciones innecesarias, pero el hecho es que la estructura de eventos en el Amstrad es tan poderosa como flexible, y tiene la ventaja adicional para un programador en código máquina de que es enteramente compatible.



El evento principal
Los eventos son el equivalente en software de las interrupciones y el bloque de eventos es empleado por el sistema operativo para llevar cuenta de todos los eventos establecidos. Siete bytes consecutivos sirven para retener la dirección de la rutina de eventos a la que se refiere el bloque, el número de eventos (si los hay) que quedan por procesar y la clase de evento

```
; Ejemplo de empleo de rutinas de temporizador del núcleo
;
;
SET.TIME:      EQU   £BD10       ;lee el contador del tiempo del reloj rapido
GET.TIME:      EQU   £BD0D       ;establece el contador del reloj rapido
;
READ.CHAR:     EQU   £BB09       ;busca un caracter del teclado
PAUSE:         EQU   300         ;pausa durante 300 impulsos rapidos
;
               LD    HL,0        ;pone a cero el tiempo
               LD    DE,0
               CALL  SET.TIME
               LD    BC, PAUSE   ;establece periodo de espera
;
;espera que un caracter se haga obtenible
;del teclado
;
WAIT:          CALL  READ.CHAR   ;esta listo el caracter?
               RET   C           ;si es asi, retrocede con acarreo
;
               CALL  GET.TIME    ;halla el tiempo
               SBC   HL,BC       ;un segundo?
                                 (o más)
;
               JR    C,WAIT      ;si no es asi, retirada
;
;si llegamos hasta aqui, la rutina ha agotado el tiempo
;vuelta con arrastre falso para señalar tiempo agotado
;
               RET
;
```

Datos básicos (X)

Con este fragmento finalizamos el detallado análisis del mapa de memoria del Commodore 64 que hemos venido realizando

ETIQUETA	DIRECCIÓN HEXA	POSICIÓN DECIMAL	DESCRIPCIÓN
ISTOP	0328-0329	808-809	Vector rutina núcleo STOP
IGETIN	032A-032B	810-811	Vector rutina núcleo GETIN
ICLALL	032C-032D	812-813	Vector rutina núcleo CLALL
USRCMD	032E-032F	814-815	Vector definido por el usuario
ILOAD	0330-0331	816-817	Vector rutina núcleo LOAD
ISAVE	0332-0333	818-819	Vector rutina núcleo SAVE
	0334-033B	820-827	No se usa
TBUFFR	033C-03FB	828-1019	Buffer Entrada/Salida cinta
	03FC-03FF	1020-1023	No se usa
VICSCN	0400-07FF	1024-2047	Área memoria pantalla de 1024 bytes
	0400-07E7	1024-2023	Matriz video: 25 líneas × 40 columnas
	07F8-07FF	2040-2047	Punteros datos sprite
	0800-9FFF	2048-40959	Espacio normal para programas BASIC
	8000-9FFF	32768-40959	ROM cartucho VSP: 8192 bytes
	A000-BFFF	40960-49151	ROM BASIC: 8192 bytes (u 8 K RAM)
	C000-CFFF	49152-53247	RAM: 4096 bytes
	D000-DFFF	53248-57343	RAM color y dispositivos Entrada/Salida o ROM generador carácter o RAM: 4096 bytes
	E000-FFFF	57344-65535	ROM núcleo: 8192 bytes (u 8 K RAM)

Editorial Delta, S.A.



10103
9 788485 822836